

سلسلة الاصول العلمية

KNA

أصول الذكاء الاصطناعي

Artificial Intelligence

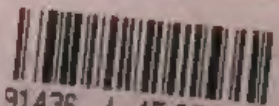
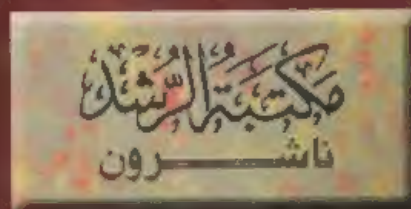


تأليف

دكتور مهندس / خالد ناصر السيد

أستاذ الحاسب الآلي المساعد

كلية العلوم - جامعة القصيم



91436 / 45 SR

COL # 193

مكتبة الملك فهد الوطنية
إيداع

سلسلة الأصول العلمية

KNA

أصول
الذكاء الاصطناعي
Artificial Intelligence



تأليف

دكتور مهندس

خالد ناصر السيد



سلسلة الأصول العلمية

KNA

أصول

الذكاء الاصطناعي

Artificial Intelligence



تأليف

دكتور مهندس

خالد ناصر السيد

فهرسك مكتبة الملك فهد الوطنية أثناء النشر

السيد ، خالد ناصر

أصول الذكاء الاصطناعي . / خالد ناصر السيد . - الرياض ١٤٢٥هـ

٢٩١ ص ، ١٧ × ٢٤ سم

ردمك : X - ٣٥٧ - ٠١ - ٩٩٦٠

١ - العنوان

١ - الذكاء الصناعي

١٤٢٥/٣٦٦١

ديوي ٠٠٦.٣

رقم الإيداع : ١٤٢٥/٣٦٦١
ردمك : X - ٣٥٧ - ٠١ - ٩٩٦٠

حقوق الطبع محفوظة

١٤٢٥هـ - ٢٠٠٤م

فروع المكتبة داخل المملكة

٢٠٥١٥٠٠ : هاتف	فرع طريق الملك فهد - غرب وزارة الشؤون البلدية والقروية	الرياض
٥٥٨٣٥٠٦-٥٥٨٤٠١ : هاتف	شارع الملك فهد مقابل مستشفى علي التويحي	عكة القرمة
٨٣٤٠٦٠٠ : هاتف	شارع أبي ذر الغفاري	عكة القرمة
٦٧٧١٣٣١ : هاتف	مقابل ميدان الطائرة	عكة القرمة
٣٢٤٢٢١٤ : هاتف	بريدة - طريق المدينة	عكة القرمة
٢٣١٧٣٠٧ : هاتف	شارع الملك فيصل	عكة القرمة
٨٧٨٢١٧٥ : هاتف	شارع ابن خلدون	عكة القرمة

وكلائنا في خارج المملكة

٢٦١٢٣٤٧ : هاتف	مكتبة الرشيد - حائل	القوي
٢٧٤٤٦٠٥ : هاتف	مكتبة الرشيد - مدينة نصر	القاهرة
٧٠١٩٧٤ : هاتف	دار ابن مسعود - حائل	بغداد
٣٠٣٦٠٩ : هاتف	المدار البيضاء / مكتبة العلم	الغروب
٨٩٠٨٨٩ : هاتف	دار الكتاب - شرقية	شمال
٦٠٣٢٥٦ : هاتف	صنعاء - دار الأمل	السبع
٩٥٧٨٣٣ : هاتف	مكتبة القرن - صنعاء	اليمن
٥٦٣٣٥٧٥ : هاتف	الشارقة - مكتبة الصحافة	الإمارات
٢٢١١١٦ : هاتف	دمشق - دار الفكر	سوريا
٤٨٦٣٥٣٣ : هاتف	مكتبة ابن القيم	قطر
٤٦٥٤٧٦١ : هاتف	عمان - دار الفكر	الأمن



مكتبة الرشيد
ناشرون

المملكة العربية السعودية
الرياض

شارع الأمير عبد الله بن عبد الرحمن
(طريق الحجاز)

س . ب : ١٧٥٢٢ - الرياض ١١٤٩٤

هاتف : ٤٥٩٢٤٥١

فاكس : ٤٥٧٢٢٨١

E-mail : alrushd@alrushdryh.com
www.rushd.com

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

قالوا سبحانك لا علم لنا إلا ما علَّمْتَنَا إِنَّكَ أَنْتَ الْعَلِيمُ الْحَكِيمُ

قَدْ جَاءَكَ مِنَ الْمُرْسَلِينَ

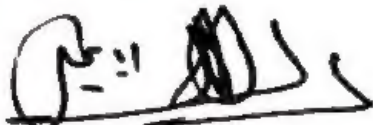
تقديم

الحمد لله رب العالمين. فقد أطلعنى الباحث و المؤلف الدكتور خالد ناصر السيد على هذه السلسلة القيمة فى علوم الحاسب الآلى. و قد جمع فيها أساسيات البرمجة و تصميم قواعد البيانات و تحليل و تصميم النظم و الذكاء الاصطناعي فى عدة لغات حديثة مهمة للباحث و المتعلم فى هذا العلم العصري و أقصد به علم الحاسب الآلى و تطبيقاته و الذى أصبح حديث المجالس و شاغل الناس.

لقد عرفت المؤلف قبل التحاقه بالسلك الأكاديمى فى كلية العلوم جامعة الملك سعود و قد ألفتته نشيطاً ومثابراً و حريصاً على تطوير نفسه و طلابه. و ما هذه السلسلة فى الأصول العلمية إلا دليلاً واضحاً على ذلك. و هى مجموعة من الكتب عرضها المؤلف بشكل مبسط يسهل على المتعلم فهم محتواها و يستفيد منها نطاق واسع من المتخصصين سواء فى علوم الحاسب أو العلوم الأخرى كالتربية أو التقنية و الطب و الزراعة و الاقتصاد و غيرها من العلوم و حتى المهتم بتطبيقاتها الأخرى.

إن كتباً مثل هذه السلسلة نحتاج إليها فى وقتنا الحاضر و فى ظل الفقر الشديد الذى تعاني منه المكتبة العربية فى المؤلفات العلمية و التطبيقية. و نحن بحاجة أيضاً و بشكل خاص لتلك المؤلفات التى تعالج و تشرح المتغير فى العلوم الحديثة كعلوم الحاسب الآلى.

و أجد لها فرصة جميلة لأشكر المؤلف الدكتور خالد ناصر السيد على جهده الموفق و بذله من وقته الثمين لإخراج هذه الكتب على وجه حديث و عاجل، لذلك أتمن له حرصه على أن تكون العلوم ميسرة و مسهلة و مذكلة بالأمثلة و التمارين و هذا سيخدم طالب الجامعة بشكل مباشر. أتمنى منه المزيد و أتمنى له المزيد من التوفيق و السداد. إنه سميع مجيب.



دكتور عبد الرحمن المهنا أبا الخيل

عميد كلية العلوم بالقصيم

تقديم

الحمد لله الذى هدانا وما كنا لنهتدى لولا أن هدانا الله و الصلاة و السلام على أشرف المرسلين وخاتم النبيين سيدنا محمد صلى الله عليه و على آله و صحبه أجمعين.

لقد منَّ الله علينا و أكرمنا و فتح علينا من فضله و رحمته ووفقنا لأن نخرج هذا الكتاب فى صورته هذه التى توفر للدارس و المتخصص منهجاً ووسيلة واضحة و مركزة و عميقة لدراسة أحدث فرع فى علوم الحاسب الآلى و هو الذكاء الاصطناعى. هذا العلم الذى أضحى يدخل فى الكثير من نواحي حياتنا و أعمالنا. لذا يوفر هذا الكتاب الفرصة للتعرف على أفرع الذكاء الاصطناعى و تطبيقات و طرق تمثيل المعرفة و البحث، وما يعرف بالمنطق الرياضى و فهم اللغات الطبيعية. نتعرف أيضاً على المعينات الإلكترونية الذكية و كذلك الخلية العصبية و النظم الخبيرة و غيرها من الموضوعات الهامة.

لقد تعرفت على الكتب العلمية التى تتوفر فى هذا المجال باللغة الإنجليزية و يكاد لا يتوفر كتاباً باللغة العربية خصيصاً للدارس الجامعى. لذلك فقد سعت مخلصاً إلى توفير هذا الكتاب ضمن سلسلة الأصول العلمية لطلبة و طالبات علوم الحاسب الآلى و نظم المعلومات فى كليات العلوم و التربية و المعلمين و التقنية. و لقد زودت الكتاب بالكثير من التطبيقات التى توفر للدارس الفرصة لكى يتمكن من الدخول بشكل جيد إلى هذا العلم الحديث.

و أتوجه بالشكر إلى سعادة الدكتور/عبد الرحمن المهنا أبا الخيل عميد كلية العلوم بالقصيم لتشجيعه لخروج هذا العمل.

وأسأل الله عز وجل أن يستقبل منى هذا العمل خالصاً لوجه تعالى

المؤلف

الفهرس

١٣	مدخل إلى الذكاء الاصطناعي	الفصل الأول
Introduction To Artificial Intelligence		
١٣	١-١ مقدمة و مفاهيم Introduction and Concepts	
١٦	٢-١ فروع الذكاء الاصطناعي AI Branches	
١٩	٣-١ تطبيقات الذكاء الاصطناعي AI Applications	
٢١	٤-١ خصائص برامج و نظم الذكاء الاصطناعي	
٢٣	٥-١ أسئلة	
٢٥	تمثيل المعرفة	الفصل الثاني
Knowledge Representation		
٢٥	١-٢ مقدمة Introduction	
٢٨	٢-٢ التمثيل بالمنطق الرياضي	
Mathematical Logical Representation		
٣١	٣-٢ المعرفة القابلة للوراثة Inheritable Knowledge	
٣١	١-٣-٢ طبيعة المعرفة القابلة للوراثة	
٣٢	٢-٣-٢ شجرة الوراثة	
٣٥	٣-٣-٢ خوارزم التوريث	
٣٦	٤-٣-٢ أمثلة محلولة باستخدام خوارزم التوريث	
٣٨	٤-٢ شبكة المعرفة اللفظية والإطار	
Semantic Net and Frame		
٣٨	١-٤-٢ شبكة المعرفة اللفظية	

٣٩	٢-٤-٢ الإطار
٤٠	المعرفة الإجرائية Procedural Knowledge ٥-٢
٤١	التمثيل باستخدام الحالات Case_Based Knowledge ٦-٢
٤٣	أسئلة ٧-٢

الفصل الثالث المعرفة الموروثة و شبكة المعرفة اللفظية ٥٩

بلغة C++

Inheritable Knowledge and Semantic Net Using C++

٤٥	Inheritance	الوراثة	١-٣
٤٨	Single Inheritance	الوراثة المفردة	٢-٣
٦٠	Multiple Inheritance	الوراثة المتعددة	٣-٣
٦٥		الاشتقاق العام و الخاص و التخلي	٤-٣
	Public, Private, and Virtual Derivation		
٦٧	Protected Variables	المتغيرات المحمية	٥-٣
٧١		الدوال التخيلية و تعدد الأشكال	٦-٣
	Virtual Functions and Polymorphism		
٧٢		تطبيق عام محلول	٧-٣
٨٢		أسئلة	٨-٣

Mathematical Logical Language (First Order Logic)

١-٤ تمثيل الحقائق و القواعد

Representing Facts and Rules

١-١-٤ قواعد التحويل إلى التمثيل الرياضى المنطقى

٢-١-٤ تمثيل الحقائق

٣-١-٤ تمثيل القواعد

٢-٤ الاستنتاج باستخدام المنطق الرياضى

Deduction using Mathematical Logic

١-٢-٤ خطوات التمثيل و الاستنتاج

٢-٢-٤ مثال ١

٣-٢-٤ مثال ٢

٤-٢-٤ مثال ٣

٥-٢-٤ تمثيل العمليات الحسابية

٦-٢-٤ مثال ٤

٣-٤ الحل باستخدام التوحيد/المطابقة و الحل

Solving using Unification and Resolution

١-٣-٤ تحويل النموذج المنطقى إلى شكل المقطع

٢-٣-٤ المطابقة/التوحيد

٣-٣-٤ الحل

٤-٣-٤ التمثيل و الاستنتاج باستخدام التوحيد و الحل

٤-٤ التمثيل المنطقى لشبكة المعرفة اللفظية

Logical Representation of Semantic Net

١-٤-٤ طريقة التمثيل

١١٥	٢-٤-٤ مثال ٨
١١٧	٥-٤ أسئلة

١٢١ معالجة اللغات الطبيعية الفصل الخامس Natural Language Processing (NLP)

١٢١	Introduction	١-٥ مقدمة
١٢٣	Language and Grammar	٢-٥ اللغات و النحو
١٢٤		١-٢-٥ وصف النحو
١٢٦		٢-٢-٥ النحو المُشكّل
١٢٨	Parsing Techniques	٣-٥ تقنيات الإعراب
١٢٨	Top-down Parsing	١-٣-٥ الإعراب من القمة لأسفل
١٢٩	Bottom-up Parsing	٢-٣-٥ الإعراب من القاع لأعلى
١٣٠	Noise Disposal Parsing	٣-٣-٥ الإعراب بإهمال الغير ضروري
١٣١	Morphology and Dictionary	٤-٥ علم الصّرف و القاموس
١٣١	Morphology	١-٤-٥ علم الصرف
١٣٢	Dictionary	٢-٤-٥ القاموس
١٣٣	Application on Context-Free Grammar	٥-٥ تطبيق لقواعد النحو الحر
١٣٣		١-٥-٥ تطبيق الإعراب طبقاً للنحو الحر
١٣٤		٢-٥-٥ مثالين للإعراب طبقاً للنحو الحر
١٣٧		٦-٥ أسئلة

الفصل السادس فهم اللغة العربية و استخدام المنطق الرياضى

Arabic Language Understanding and Using Mathematical Logic

١٤١	Preface	تمهيد	١-٦
١٤٢		نظم معالجة اللغات الطبيعية	٢-٦
	Natural Language Processing System		
١٤٣		١-٢-٦ مهام نظم معالجة اللغة الطبيعية	
١٤٣		٢-٢-٦ مكونات نظم معالجة اللغة الطبيعية	
١٤٤		٣-٢-٦ مراحل إنشاء نظم معالجة اللغة الطبيعية	
١٤٦		نظام استعلام باللغة العربية الفصحى	٣-٦
	Arabic Question Answering System (AQAS)		
١٤٦		١-٣-٦ بناء نظام AQAS	
١٥٠		٢-٣-٦ أنماط و وظائف نظام AQAS	
١٥١		٣-٣-٦ مجال المعرفة في نظام AQAS	
١٥١	Design of AQAS System	تصميم نظام AQAS	٤-٦
١٥٢		١-٤-٦ لغة الاستعلام في نظام AQAS (المدخلات النصية)	
١٥٣		٢-٤-٦ النحو في نظام AQAS	
١٥٦		٣-٤-٦ الإعراب (Parsing)	
١٥٧		القاموس و عملية الصرف اللغوى	٥-٦
	Dictionary and Morphological Process		
١٥٧		١-٥-٦ القاموس	
١٥٩		٢-٥-٦ التشكل/التصريف اللغوى	
١٦٠	Knowledge Representation	تمثيل المعرفة	٦-٦
١٦١		١-٦-٦ النظام المبني على الإطار	

١٦٢	٢-٦-٦ هيكل الفتحة	
١٦٢	٣-٦-٦ هيكل الإطار	
١٦٣	٧-٦ مثال تنفيذ استعلام	Executing a Query
١٦٥	٨-٦ أسئلة	

١٦٧ نظم المنطق الغامض Fuzzy Logic Systems الفصل السابع

١٦٧	١-٧ تمهيد	Preface
١٦٩	٢-٧ مفهوم المنطق الغامض	Fuzzy Logic Concept
١٦٩	١-٢-٧ الفكرة المركزية لنظم المنطق الغامض	
١٧٠	٢-٢-٧ مميزات نظام المنطق الغامض	
١٧٢	٣-٢-٧ أهمية نظم المنطق الغامض	
١٧٤	٤-٢-٧ استخدام نظم المنطق الغامض	
١٧٥	٣-٧ مُتَحَكِّمٌ بالمنطق الغامض لعمود تقطير	Fuzzy Logic Controller for a Distillation Column
١٧٦	١-٣-٧ تقديم للتطبيق	
١٧٧	٢-٣-٧ وصف التحكم بالعملية	
١٧٩	٣-٣-٧ المتحكم بالمنطق الغامض	
١٨٤	٤-٧ استخدام المنطق الغامض مع أوامر SQL	Using Fuzzy SQL Commands
١٨٧	٥-٧ أسئلة	

Expert Systems

١٨٩	Introduction	١-٨	مقدمة
١٨٩		١-١-٨	مفهوم النظم الخبيرة
١٩٠		٢-١-٨	مميزات النظم الخبيرة
١٩١		٣-١-٨	تطبيقات النظم الخبيرة
١٩٢		٢-٨	تطوير النظم الخبيرة
	Expert Systems Development		
١٩٢		١-٢-٨	العاملون في النظم الخبيرة
١٩٣		٢-٢-٨	خطوات إنشاء النظم الخبيرة
١٩٤		٣-٢-٨	أمثلة من النظم الخبيرة
١٩٦		٣-٨	أداة التصنيف و التشخيص : ناصر ٩٦
	Classification and Diagnosing Tool : Nasser96		
١٩٧		٤-٨	وصف الأداة التصنيف ناصر ٩٦
	Overview of Nasser96 Tool		
١٩٩		١-٤-٨	تركيب أداة التصنيف
٢٠١		٢-٤-٨	وصف الحالة المدخلة
٢٠٣		٣-٤-٨	الحدود و القيود في أداة ناصر ٩٦
٢٠٥		٥-٨	المعرفة الممثلة في الأداة ناصر ٩٦
	Knowledge Represented by NASSER96		
٢٠٥		١-٥-٨	تركيب شبكة المعرفة اللفظية
٢٠٦		٢-٥-٨	جلب و تمثيل التفسير
٢٠٧		٦-٨	التصنيف و الاستدلال على الحل باستخدام الحالات
	Classification & Case_Based Reasoning		
٢٠٨		١-٦-٨	الاستدلال على الحل

٢١٠	٢-٦-٨ التصنيف
٢١٣	٧-٨ تطبيقات الأداة Tool Applications
٢١٣	١-٧-٨ تشخيص أمراض الحساسية
٢١٤	٢-٧-٨ تصنيف حشرات المزارع
٢١٦	٨-٨ أسئلة

٢١٧ الشبكات العصبية الفصل التاسع

Neural Networks

٢١٧	١-٩ مقدمة Introduction
٢١٧	١-١-٩ مفهوم الشبكة العصبية البشرية
٢١٩	٢-١-٩ العقل البشري نظام معالجة معلومات
٢٢٠	٣-١-٩ مفهوم الشبكات العصبية الاصطناعية
٢٢٢	٤-١-٩ الشبكات العصبية الاصطناعية مقابل البرامج التقليدية
٢٢٣	٢-٩ استخدام الخلايا العصبية الاصطناعية

Using Artificial Neurons

٢٢٣	١-٢-٩ خلية عصبية بسيطة
٢٢٤	٢-٢-٩ قواعد التنفيذ
٢٢٦	٣-٢-٩ خلية عصبية أكثر تقدماً
٢٢٧	٣-٩ عمارة الشبكات العصبية

Architecture of Neural Networks

٢٢٧	١-٣-٩ شبكات التغذية الأمامية Feed-Forward Network
٢٢٨	٢-٣-٩ شبكات التغذية العكسية Feed-Forward Network
٢٢٩	٣-٣-٩ طبقات الشبكة Network Layers
٢٣٠	٤-٩ التعلم Learning

٢٣٠	١-٤-٩ تصنيف الشبكات
٢٣٠	٢-٤-٩ تصنيف التعلم
٢٣١	٣-٤-٩ سلوك الشبكة العصبية الاصطناعية
٢٣٢	٤-٤-٩ قاعدة Hebb للتعلم
٢٣٣	٥-٩ تطبيقات الشبكات العصبية الاصطناعية
	Applications of Artificial Neural Networks
٢٣٤	٦-٩ أسئلة

٢٣٧ المعينات الإلكترونية و التعليم **Agents & Teaching** الفصل العاشر

٢٣٧	١-١٠ مقدمة	Introduction
٢٣٨	٢-١٠ طبيعة المعين الذكي	
		The Nature of Intelligent Agent
٢٣٨	١-٢-١٠ معيار الكفاءة للمعين	
٢٤٠	٢-٢-١٠ تصميم المعين	
٢٤١	٣-٢-١٠ بيانات المعين	
٢٤٢	٤-٢-١٠ خصائص المعين الذكي	
٢٤٤	٥-٢-١٠ المعين مُعَلِّم و مُتَعَلِّم	
٢٤٤	٣-١٠ تطبيقات المعين الذكي	
		Intelligent Agent Applications
٢٤٤	١-٣-١٠ معينات تعليمية	
٢٤٥	٢-٣-١٠ معينات واجهة اتصال ذكية	
٢٤٦	٣-٣-١٠ معينات تطوير	
٢٤٧	٤-١٠ تركيب المعين	Agent Structure

٢٤٧	١-٤-١٠ دالة النموذج العام للمعين
٢٤٩	٢-٤-١٠ معين جدول المتابعة
٢٥٠	٢-٤-١٠ معين يدرك ما حوله من العالم
٢٥٢	٥-١٠ الاتجاهات و التطويرات في تقنية المعينات
٢٥٥	٦-١٠ المعين المرشد للمتدربين على لغة C++ An Advisor Agent for C++ Trainees
٢٥٥	١-٦-١٠ المعين المرشد
٢٥٦	٢-٦-١٠ تركيب المعين
٢٥٩	٣-٦-١٠ أداء المعين
٢٦٠	٧-١٠ أسئلة

٢٦١	البحث و تَعَلُّم الآلة	الفصل الحادي عشر
	Search and Machine Learning	

٢٦١	Search Techniques	١-١١	تقنيات البحث
٢٦١	Search Space/State Space	١-١-١١	١-١-١١ حيز البحث/فضاء الحالات
٢٦٤	Example of Disk Search Problem	٢-١-١١	٢-١-١١ مثال مشكلة بحث في القرص عن ملف معين
٢٦٤	Uninformed (Blind) Search	٣-١-١١	٣-١-١١ البحث الأعمى
٢٦٥	Depth-First Search		(أ) البحث العمقى
٢٦٧	Breadth-First Search		(ب) البحث العرضى
٢٦٩	Parsing as a Search Problem	٤-١-١١	٤-١-١١ الإعراب كمسكلة بحث
٢٧١	Informed (Heuristic) Search	٥-١-١١	٥-١-١١ البحث التجريبي الموجّه
٢٧١	Machine Learning	٢-١١٠	تَعَلُّم الآلة

٢٧٢ Learning by Being Told ١-٢-١١ التعلّم الإملائي

٢٧٣ ٢-٢-١١ التعلّم المبني على التشابه

Similarity-Based Learning

٢٧٤ ٣-٢-١١ التعلّم المبني على التفسير

Explanation-Based Learning

٢٧٤ ٤-٢-١١ التعلّم المبني على الحالات

Case-Based Learning

٢٧٥ ٥-٢-١١ التعلّم الاستقرائي (تحت إشراف)

Inductive (Supervised) Learning

٢٧٨ ٣-١١ أسئلة

٢٧٩ التعرف على الكلام و توليده الفصل الثاني

Speech Recognition and Synthesis

عشر

٢٧٩ Preface تمهيد ١-١٢

٢٨٠ Phonemes الأصوات اللغوية ٢-١٢

٢٨٢ التعرف على الكلام (سماع الصوت) ٣-١٢

Speech Recognition

٢٨٣ Speech Synthesis توليد الكلام (إصدار الصوت) ٤-١٢

٢٨٥ ٥-١٢ أسئلة

٢٨٧ References المراجع

الفصل الأول

مدخل إلى الذكاء الاصطناعي

Introduction to Artificial Intelligence

منذ فترة غير بعيدة بزغ علم جديد من علوم الحاسب الآلى ذاع صيته و هيمن على أغلب أبحاث الحاسب الآلى. و قد تعددت فروع هذا العلم و كثرت تطبيقاته و امتدت لتشمل الكثير من نواحي الحياة. نشأت أيضاً معاهد و مجلات علمية عديدة كل منها متخصصة فقط فى أحد فروع هذا العلم. وفى الفترة الأخيرة بدأت بعض ثمار هذا العلم تأخذ موقعها و تخرج فى ثوب تجارى تفيد الإنسان فى حياته العملية. هذا العلم يُصنّف على أنه الجيل الخامس من أجيال برمجيات الحاسب الآلى الذى صاحبه ظهور العديد من لغات برمجة خاصة به هى لغات الجيل الخامس (Fifth Generation Languages). العلم النشط الهام و المتطور الذى نتحدث عنه هو علم الحاضر و المستقبل و محور هذا الكتاب هو علم الذكاء الاصطناعي (Artificial Intelligence (AI).

١-١ مقدمة و مفاهيم Introduction and Concepts

الذكاء (Intelligence) هو الجزء الحساس من القدرة على تحقيقه الأهداف أو الوصول إليها وتختلف درجاته وأنواعه بين البشر و الحيوانات والآلات. فى البداية

كان الهدف من برامج الذكاء أن تحل محل الخبير في تخصص البرامج ولكن ثبت استحالة ذلك وأصبح الهدف من برامج الذكاء هو مساعدة الخبير في أداء عمله بسرعة و كفاءة متميزة.

الذكاء الاصطناعي (Artificial Intelligence (AI) هو العلم والهندسة اللذان يجعلان الحاسب الآلي آلة ذكية وهو اصطناعي لأنه عبارة عن برامج وأجهزة تتعاون لتؤدي عملية فهم معقدة يمكن أن تضاهي ذكاء البشر من فهم وسمع ورؤية وشم وكلام وتفكير. أي أنه برامج ذكية + أجهزة = ذكاء اصطناعي.

نستطيع القول أيضاً بأن الذكاء الاصطناعي هو المساحة من علم الحاسب التي تركز على إنشاء آلة تستطيع القيام بما يقوم به الإنسان من أعمال ذكية. و الحلم في تطوير آلة ذكية أصبح قريب المنال.

بدأ موضوع الذكاء الاصطناعي عام ١٩٤٧ م على يد العالم الآن تيورنج Alan Turing. و لقد حدد هذا العالم أن الذكاء الاصطناعي هو عمل برامج ذكية و ليس بناء آلات ذكية. الذكاء اصطناعي يجعل الآلة (أي الحاسب) تبدو وكأنها ذكية و لكن الواقع أن الآلة كما هي لم تتغير و لكن مطور البرنامج هو الذي أعد البرنامج بحيث يؤدي أعمال ذكية أو لنقل تبدو ذكية.

اختبار تيورنج (Turing Test) : حدد تيورنج أن الآلة الذكية هي التي تنجح في أن تبدو كالإنسان في ملاحظتها للمعرفة.

الآلة الطفلة (Child Machine) : هي الآلة التي يمكنها أن تتحسن بالتعلم من الخبرة. و قد طُرِحت هذه الفكرة عدة مرات ولكن لم تصل برامج الذكاء الاصطناعي إلى المستوى الذي يمكننا من تحقيق ذلك.

هل هناك ضرورة لبرامج AI (الجيل الخامس)؟

نعم هناك ضرورة لأن البرامج واللغات التي سبقت ظهوره أو حتى تلت ذلك (الجيل الرابع وما قبله) غير قادرة على أداء الأهداف التي من أجلها نشأت برامج الذكاء الاصطناعي. وهناك برامج ونظريات وتجارب تتم حالياً بعضها وصل إلى نتائج جيدة والبعض الآخر في الطريق إلى ذلك. تشتعل الأبحاث و تتلاحق في مختلف أفرع و تطبيقات الذكاء الاصطناعي لتخرج لنا كل يوم بما هو جديد و بناءً.

هل الذكاء الاصطناعي مُضاهٍ أو مُشابه لذكاء الإنسان؟
أحياناً بنسبة مُعينة و ليس دائماً.

هل الذكاء الاصطناعي ينوى وضع عقل الإنسان داخل الحاسب الآلي؟
في البداية أقر بعض الباحثين أن هذا هو الهدف و لكن ذلك بعيد المنال لأن عقل الإنسان ملئ بالغرائب و الصعوبات و غير مفهوم حتى الآن بصورة كاملة.

ما العلوم الواجب دراستها قبل الذكاء الاصطناعي؟

- الرياضيات ← الرياضيات المنطقية.
- أحد لغات البرمجة الخاصة بالذكاء الاصطناعي ← لغة Java أو C++ أو Prolog.

- بعض التطبيقات ← اللغات أو علم النفس أو علم الأحياء أو غير ذلك.

من المفيد أن نعلم أن كافة برامج و نظم الذكاء الاصطناعي تحاول أن تضاهي حواس و خصائص الإنسان المختلفة مثل الحركة محسوبة و التفكير و الشم و الرؤية و التعرف على الأشكال و النماذج المعقدة والتدقيق و التكلم و إدراك الكلمات و فهم معانيها و غير ذلك. أحياناً يتفوق أداء برنامج الذكاء الاصطناعي

على الإنسان و لكن ذلك في السرعة فقط أما الإنسان الخبير فلا شك أنه أقدر من البرنامج في بعض النواحي.

إن الذكاء الاصطناعي يعتبر من أهم علوم الحاسب الآلي خاصة في الوقت الحاضر. فنحن بحاجة ماسة لمعرفة والإلمام به. هذا العلم يعتبر أساس التقنيات الحديثة والتي نراها الآن في المجال العسكري وفي المجال المدني في كثير من الاستخدامات. سوف ندرس في هذا الفصل بعض أفرع الذكاء الاصطناعي و تطبيقاته و طرق البحث الخاصة به.

AI Branches

١-٢ فروع الذكاء الاصطناعي

تعددت المجالات التي شملها علم الذكاء الاصطناعي لدرجة أن المتخصص في هذا العلم أصبح متخصصاً فقط في فرع واحد فقط أو فرعين من فروع و ليس في كافة فروع هذا العلم. فيما يلي نسرد بعضاً من تلك الفروع لنتعرف على بعض المجالات التي تقع بين جنبات هذا العلم.

• الذكاء الاصطناعي المنطقي (Logical AI)

هي البرامج التي تعرف حقائق عن تطبيق معين والأهداف المطلوب تحقيقها ويتم تمثيل الحقائق والأهداف بلغة المنطق الرياضي (Mathematical Logical language).

• البحث (Search)

جميع البرامج تبحث في عدد كبير جداً من الاحتمالات (مثل الحركة في لعبة الشطرنج) وذلك لإيجاد حل لأحد المشاكل. يهدف AI إلى جعل البحث أكثر سرعة ودقة. و توجد طرق عديدة للبحث نتعرف على بعضها في هذا الكتاب.

• **ملاحظة الشكل أو النموذج (Pattern Recognition)**

هي برامج تستطيع ملاحظة (التعرف) الأشكال و النماذج والتمييز بينها أكثر من دقة العين. على سبيل المثال : التعرف على صاحب البصمة و قراءة خط اليد بلغة ما.

• **تمثيل المعرفة (Knowledge Representation)**

يتم تمثيل حقائق عن أحد مجالات الحياة (مجال التطبيق) بأحد طرق تمثيل المعرفة مثل لغة المنطق الرياضى أو النموذج (Model) أو الإطار (Frame) أو القواعد (Rule_based) و شبكة المعرفة (Semantic Net) و التمثيل بالحالات (Case_based) بالإضافة إلى طرق أخرى. فى الفصل الثانى نتعرف على طرق تمثيل المعرفة بمزيد من التفصيل.

• **الاستنتاج (Inference)**

هو استنتاج معرفة جديدة من معرفة قديمة أى استخلاص معلومات للوصول إلى الحل. و تُعرف عملية الاستدلال على الحل باسم Reasoning.

• **التعلم من الخبرة (Learning From Experience)**

يمكن للبرامج أن تتعلم بعض الحقائق الجديدة أو الإجراءات الممكن اتخاذها فى المواقف الجديدة القريبة الصلة من مواقف سابقة أو تعلم قاعدة (قانون معين) من مثال.

• التخطيط (Planning)

هو برامج تبدأ بحقائق عامة من مجال معين وخصوصاً حقائق عن تأثير الإجراءات وكذلك الأهداف المطلوبة . تقوم هذه البرامج بتوليد (إنشاء) خطة استراتيجية للوصول إلى الهدف. (الخطة هي سلسلة من الخطوات أو الإجراءات).

• المعينات الإلكترونية الذكية (Artificial Agent)

هي برامج تشبه الدوال تقوم بمراقبة (ملاحظة) البيئة التي تعمل عليها وعند قياس (حدوث) شئ معين تقوم باتخاذ الإجراء المناسب فوراً. وتوجد بعض المعينات التي تحتفظ بخلفية سابقة عن البيئة وتنتظر حدوث سلسلة من الأحداث لتقوم باتخاذ الإجراء المناسب

• تقنيات المنطق الغامض/المشوش (Fuzzy Logic Technique)

هي برامج/أساليب تستخدم قواعد المنطق المشوش fuzzy-logic في تحليل الأشياء وذلك لصعوبة استخدام الإجابة (نعم - لا). لتحديد بعض الحلول ولكن يمكن استخدام مصطلحات تعبر بشكل اقرب لفهم البشر وهي على سبيل المثال (ممتاز - جيد جداً - جيد - ضعيف).

• شبكة الخلايا العصبية والحسابات المتوازية

(Neural Network & Parallel Computations)

عقل إنسان يتكون من شبكة (مثل شبكة الإنترنت) من بلايين الخلايا العصبية، كل خلية تسمى نيورون (Neuron). كل خلية بمفردها لا تستطيع أن تؤدي عملاً ذو معنى. لكن كل مجموعة من الخلايا يمكن أن تتعاون لتؤدي عملاً معيناً في نفس الوقت الذي تتعاون فيه مجموعات أخرى لتؤدي أعمالاً أخرى. يتم عمل برامج تستخدم نفس هذه الفكرة بإنشاء وحدات تشبه

النيورون وتقوم في مجملها بمجموعة من الحسابات المتوازية من خلال عمليات رياضية منطقية.

• الخوارزميات الجينية (Genetic Algorithms(GA))

اكتشف علماء الطب الجينات الطبيعية و تعرفوا على تركيبها و اكتشفوا أن الجينات الطبيعية تتحكم في خصائصنا الطبيعية و أدائنا و شخصيتنا و صحتنا. و لقد قاد هذا الاكتشاف الرائع إلى استخدام الخوارزميات الجينية (GA) في حسابات التطبيقات الهندسية. تتضمن الخوارزميات مجموعة من العمليات العامة التي تتكامل لحل الكثير من المشاكل الدقيقة أو الغير ملموسة. الخوارزميات الجينية هي طريقة بحث مبنية على ما يُطلق عليه تَجَمُّع مجموعة الحلول (population) و هو مجموعة من نقاط عينات توفر معلومات عن عدة مستويات من الخصائص.

٣-١ تطبيقات الذكاء الاصطناعي AI Applications

أيضاً تنوعت و تشعبت تطبيقات الذكاء الاصطناعي لتضم العديد من التطبيقات. من خلال هذه التطبيقات يستطيع مطورو نظم و برامج الذكاء الاصطناعي من تطبيق أكثر من فرع في تطبيقاتهم. فيما يلي نسرّد بعضاً من التطبيقات الهامة في مجال الذكاء الاصطناعي.

- الألعاب الذكية (Intelligence Games) : مثل لعبة الشطرنج (Chess game).

• تمييز الكلام (Speech Recognition)

هى برامج تستطيع تحويل الأصوات إلى كلمات (text) على الحاسب. وهناك برامج تمكن المستخدم من توجيه أوامر وجمل للحاسب (السكرتير الآلى). بعض الأماكن ذات الوضع الأمنى المميز تستخدم الصوت للتعرف على الموظفين أو العملاء فى البنوك.

• صناعة الكلام (Speech Synthesis)

هى برامج تستطيع تحويل الكلمات و الجمل المكتوبة على الحاسب (text) إلى أصوات. وهناك برامج تمكن المستخدم من قراءة الجمل و ترجمتها و هى تفيد جميع المستخدمين و خصوصاً ذوى الإعاقة البصرية أو اليدوية.

• تمييز و قراءة الحروف (Character Recognition)

هى برامج تستطيع قراءة حروف و كلمات مكتوبة باليد أو مطبوعة و تحويلها إلى حروف و كلمات و جمل على الحاسب (text). بعد ذلك نستطيع استخدام هذا النص كما لو كنا قد أدخلناه من لوحة المفاتيح.

• تمييز النماذج و التعرف عليها (Pattern Recognition)

هى برامج تستطيع التعرف على النماذج المختلفة مثل التعرف على بصمة اليد (Finger Print) و غيرها.

• فهم اللغات الطبيعية (Natural Language Understanding)

برامج تمكن الحاسب من فهم لغة طبيعية (مكتوبة أى Text) مثل اللغة العربية أو الإنجليزية أو أى لغة أخرى فى مجال تطبيق معين. و نعى بالفهم هنا هو التعرف أولاً على التركيب النحوى للجمل و موقع كل كلمة من الإعراب ثم فهم معنى الجملة و الرد عليها سواء بإضافة معلومة جديدة

إلى قاعدة المعرفة أو استخراج معلومة معينة مطلوبة من قاعدة المعرفة أو التحقق من صحة معلومة من عدمه. مثال على ذلك نظم AQAS و Eliza.

• الرؤية بالحاسب (Computer Vision)

هي برامج تُستخدم في التعرف على الصور و الكائنات. من ضمن هذه البرامج : برامج تستخدم مع الذراع الآلى للتعرف على الأجزاء المختلفة قبل نقلها أو تركيبها و التحرك داخل مفاعل نرى أو إجراء لحام خطر تحت الماء أو تجميع الأجهزة الدقيقة بالإضافة إلى العديد من التطبيقات.

• نظم الخبرة (Expert System)

هي برامج تُستخدم لاستخلاص و تجميع خبرة العديد من الخبراء من مجال معين ثم تستخدم لإيجاد حلول للمشاكل المماثلة في الطب و الزراعة والاقتصاد. أمثلة على ذلك : نظم Mycin و Nasser96.

• الكائنات الآلية أو الأذرع الآلية (Robotics)

هي أجهزة يتم تصميمها و تطويرها بحيث تؤدي بعض التحركات و أحياناً بعض الأصوات و تستطيع حمل بعض المواد الخطرة أو الدخول في منطقة إشعاعية أو الكشف عن متفجرات أو اللحام تحت الماء أو تجميع الأجهزة الإلكترونية الدقيقة و السيارات. تستخدم هذه الآلات الكثير من تقنيات الذكاء الاصطناعي مثل الرؤية بالحاسب و التفكير و تمييز و نطق الكلام.

٤-١ خصائص برامج و نظم الذكاء الاصطناعي

Characteristics of AI Programs and Systems

البرامج و النظم التي نبنيها و تقع تحت تصنيف برامج الذكاء الاصطناعي لابد و أنها تختلف عن البرامج التقليدية و نظم قواعد البيانات. فهذه الفئة من البرامج قد

أُعِدَّتْ لكي تؤدي وظيفة لا يمكن أن تؤديها البرامج و النظم التقليدية. لأن برامج الذكاء الاصطناعي قد نشأت بالأساس لتضاهي حواس الإنسان.

وقد استطاع الباحثون إنشاء برامج و نظم تقارب بصورة مُصَغَّرَة تفكير الإنسان مثل فهم الكلام و الطيار الآلي و نظم الفضاء و النظم العسكرية. هذا بالإضافة إلى التطبيقات التجارية التي بدأت تغزو الأسواق مثل الغسالة الذكية (Fuzzy Washing Machine) التي تحدد درجة الحرارة المطلوبة و كمية مسحوق الغسيل اللازم و كمية المياه و سرعة دوران المحرك حسب نوع القماش و درجة الاتساخ و كمية الملابس المطلوب غسلها و كذلك الغرامل الذكية في السيارات و السخانات الذكية في المصانع و أعمدة تكرير البترول.

الصفات الواجب توافرها في برامج الذكاء الاصطناعي هي :

- **وافية التمثيل (Representation Adequacy)**
يجب أن يكون البرنامج قادراً على تمثيل كل أنواع المعرفة المرتبطة بمجال تطبيق معين.
- **وافية الاستدلال (Inferential Adequacy)**
يجب أن يكون البرنامج لديه القدرة على معالجة تركيب التمثيل لاشتقاق تراكيب جديدة لتتوافق مع المعرفة الجديدة المشتقة من المعرفة القديمة و إضافتها إلى قاعدة المعرفة الخاصة به.
- **كفاءة الاستدلال (Inferential Efficiency)**
يجب أن يكون البرنامج لديه القدرة على دمج معلومات جديدة في تراكيب المعرفة لتوجيه برنامج الاستدلال إلى الحل.

• كفاءة الاستنباط (Acquisitional Efficiency)

يجب أن يكون البرنامج لديه القدرة على أخذ (استنباط/استخلاص) معلومات جديدة من المستخدم أو البيئة والتحكم في إدخال المعلومات.

١-٥ أسئلة

١. عرف Intelligence و Artificial Intelligence.
٢. ما المقصود بالآلة الطفلة و هل نجحت؟
٣. ما هو Turing Test؟
٤. أذكر بعض أفرع الذكاء الاصطناعي؟
٥. أذكر بعض تطبيقات الذكاء الاصطناعي؟
٦. ما المقصود بكل من : Expert systems و Pattern Recognitions و Computer Vision و Robotics و Speech Synthesis & Recognition.
٧. ما هي الشروط الواجب توافرها في برنامج الذكاء الاصطناعي.

الفصل الثاني

تمثيل المعرفة

Knowledge Representation

يلعب تمثيل المعرفة (Knowledge Representation (KR)) في قاعدة معرفة (Knowledge Base) الدور الأكبر و الأهم في برامج الذكاء الاصطناعي لأن برامج الذكاء الاصطناعي هي في الأساس عبارة عن طريقة تمثيل للمعرفة و طريقة الحصول على المعرفة. على قدر النجاح في اختيار الطريقة الأنسب و استخدامها في تمثيل جيد للمعرفة في مجال تطبيق معين بقدر نجاح برنامج الذكاء الاصطناعي.

Introduction

١-٢ مقدمة

لحل المشاكل المعقدة تحتاج برامج الذكاء الاصطناعي كمية كبيرة من المعرفة وطرقاً مميزة لتناول تلك المعرفة. عملية تمثيل المعرفة (Knowledge Representation) هي عملية تمثيل للمعرفة باستخدام أسلوب معين و ذلك بتحويلها من الشكل التقليدي المعروف لنا إلى تمثيل داخلي يستطيع برنامج AI معالجتها و الاستفادة منها بأشكال متعددة حسب نوع التطبيق و الغرض منه.

تتعدد طرق تمثيل المعرفة، لكن جميع الطرق تكون مرتبطة بمجال التطبيق (domain-specific knowledge). لذلك تسمح النماذج الدقيقة لتمثيل المعرفة

لتقنيات حل المشاكل (problem-solving) بالعمل عليها وإيجاد حلول لما يُستَجَد من مشاكل.

يتضمن كل مجال تطبيق -جزء من العالم الواقعي يمكن تسميته العالم المُصَغَّر الخاص بالتطبيق- حقائق معينة و هي الأشياء المطلوب تمثيلها باستخدام طريقة معينة. هذه الأشياء هي التي سوف يتناولها البرنامج و مستخدميه.

تأمل الجمل (الحقائق) الموجودة في شكل ١-٢. هذه الجمل سواء كانت باللغة العربية أو اللغة الإنجليزية (أو بأي لغة أخرى) فإنها تُعَد تمثيل للمعرفة بأحد اللغات الطبيعية التي يفهما الإنسان.

محمد مصري

فهد سعودي

جميع السعوديون عرب

يجب على كل مسلم أن يهتم بالقضايا الإسلامية

Mohammad is Egyptian.

Fahd is Saudi.

All Saudis are Arabian.

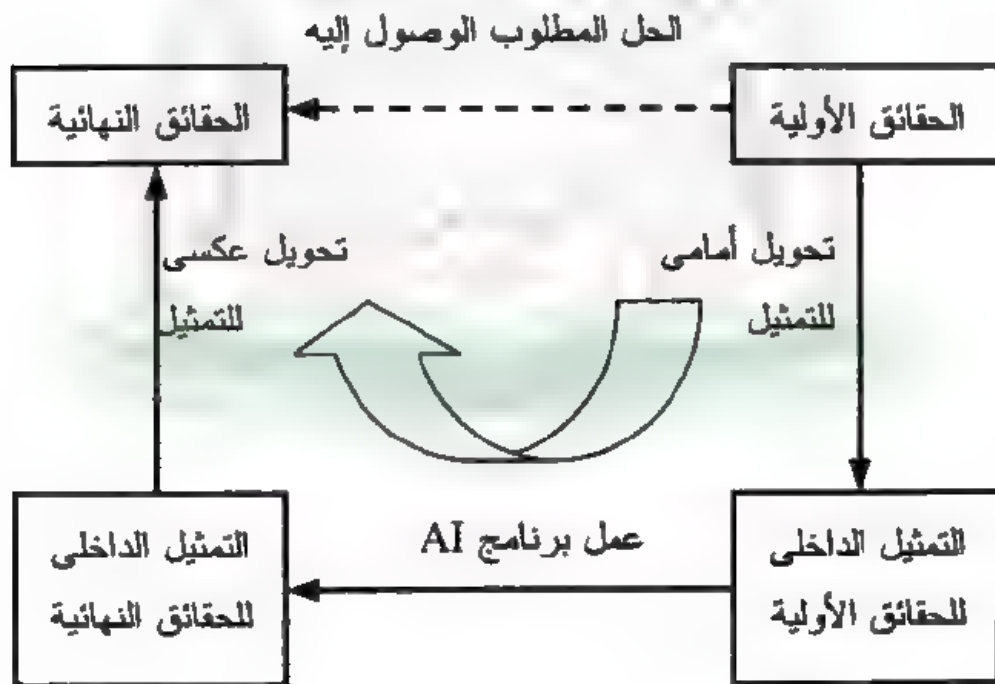
Any Moslem should care Islam issues.

شكل ١-٢ : تمثيل الحقائق باللغتين العربية و الإنجليزية.

هذه الجمل بشكلها الطبيعي هذا لا يستطيع برنامج AI العمل عليها و لابد من تحويلها إلى تمثيل آخر بحيث يستطيع برنامج AI معالجته و اشتقاق أو استنتاج معرفة جديدة منه أو استخدام لإيجاد حلول لمشاكل تطبيق معين. نستطيع تمثيل تلك الحقائق بأكثر من طريقة. على سبيل المثال، نُمثِّل الجملتين الأوليين من الجمل الإنجليزية بالتمثيل الرياضي المنطقي كما يلي :

Egyptian (Mohammad).
Saudi (Fahd).

هنا التمثيل الرياضي المنطقي هو الطريقة المستخدمة في تمثيل المعرفة و يُطلق عليها التمثيل الداخلي (Internal Representation) الذي يضعه البرنامج في ذاكرة الحاسب ليستخدمه برنامج AI. أما التمثيل اللغوي الموجود في شكل ٢-١ فهو التمثيل الخارجي (External Representation) الذي يفهمه الإنسان و هو خارج البرنامج. شكل ٢-٢ يوضح عملية تمثيل الحقائق و تحويلها من التمثيل الخارجي إلى التمثيل الداخلي و عمل برنامج AI.



شكل ٢-٢ : تمثل الحقائق و تحويلها و دور برنامج AI.

كما نرى في شكل ٢-٢ أن برنامج AI يعمل على التمثيل الداخلي للحقائق الأولية ليصل إلى التمثيل الداخلي للحقائق النهائية (الهدف المطلوب الوصول إليه) أي الحل. للوصول إلى التمثيل الداخلي للحقائق الأولية يجب إجراء تحويل أمامي

للتمثيل الخارجى (forward representation mapping) للحقائق الأولية. و للوصول إلى التمثيل الخارجى للحقائق النهائية التى وصل إليها برنامج AI يجب إجراء تحويل عكسى للتمثيل الداخلى (backward representation mapping) للحقائق النهائية.

تتعدد طرق تمثيل المعرفة و كذلك أنواع المعرفة الممكن تمثيلها فى مجال الذكاء الاصطناعى. فهناك تمثيل للمعرفة باستخدام المنطق الرياضى أو باستخدام التمثيل بالحالات أو المعرفة القابلة للوراثة أو المعرفة الإجرائية أو التمثيل باستخدام الإطار من بين طرق تمثيل المعرفة. البنية المصممة يجب أن تتواءم مع تقنية لاستنتاج المعرفة المطلوب استنتاجها.

من الممكن أن نستخدم طريقة أو أكثر من هذه الطرق فى تمثيل المعرفة فى تطبيق واحد. فى الجزء الباقى من هذا الفصل نتعرف على بعض هذه الطرق و فى الفصول التالية نجد العديد من التطبيقات حول بعض هذه الطرق.

٢-٢ التمثيل بالمنطق الرياضى

Mathematical Logical Representation

تمثيل المعرفة باستخدام المنطق الرياضى أو ما يُعرف بلغة المنطق الرياضى (Mathematical Logical Representation) يعتمد على أن المعرفة تأخذ أحد شكلين هما :

- **الحقائق (Facts)** و هى معلومة تصف شئ بذاته وتخصه (اسمية أو فعلية أو شبه جملة). على سبيل المثال، تأمل الجمل الموجودة فى شكل ٢-٣ و تعتبر حقائق.

- القواعد (Rules) و هي معلومة يمكن تعميمها أو تطبيقها على مجموعة من الأشياء مثل (من يذاكر ينجح) ويلزم لتطبيقها أو تعميمها توفر شرط أو مجموعة شروط. على سبيل المثال، تأمل الجمل الموجودة في شكل ٢-٤ و تعتبر قواعد.

عزة مسلمة.

سيف يتكلم العربية.

الذكاء الاصطناعي مقرر حاسب.

اللقط حيوان.

المانجو طعام.

Azzah is a Moslem.

Sayf speaks Arabic.

Artificial intelligence is a computer course.

Mango is a food.

شكل ٢-٣ : أمثلة لجمل لغوية تمثل حقائق.

أحمد يحب جميع مقررات الحاسب الآلي.

جميع العرب يتكلمون العربية.

آلاء تأكل أى شيء تأكله دعاء.

طلاب الحاسب الآلي يدرسون لغات البرمجة.

Ahmad likes all computer courses.

All Arab speak Arabic.

Alla eats anything Doaa eats.

All Computer students study programming languages.

شكل ٢-٤ : أمثلة لجمل لغوية تمثل قواعد.

يتم تمثيل الحقائق و القواعد في لغة المنطق الرياضى باستخدام المُسند

(predicate). و الشكل العام للمسند هو :

Predicate_Name (Argumen 1, ..., Argument_n)

حيث اسم المسند هو Predicate name و العناصر بين القوسين arguments هى المتغيرات و الثوابت الذين يشملهم المسند و نستخدم المسند فى تمثيل الحقيقة بينما يمكن أن يتم تمثيل قاعدة معينة باستخدام مسندين أو أكثر. شكل ٢-٥ يبين التمثيل الرياضى المنطقى للحقائق الموجودة فى شكل ٢-٣. بينما يبين شكل ٢-٦ التمثيل الرياضى المنطقى للقواعد الموجودة فى شكل ٢-٤.

moslem(Azzah).

speaks(Sayf,Arabic).

computer_course(Artificial_intelligence).

food(Mango).

شكل ٢-٥ : التمثيل الرياضى المنطقى للحقائق

الموجودة فى شكل ٢-٣.

$\forall x: \text{computer_course}(x) \rightarrow \text{likes}(\text{Ahmad}, x).$

$\forall x: \text{Arab}(x) \rightarrow \text{speak}(x, \text{Arabic}).$

$\forall x: \text{eats}(\text{Doaa}, x) \rightarrow \text{Alla eats anything Doaa eats}.$

$\forall x: \forall y: \text{computer_student}(x) \wedge \text{programming_language}(y) \rightarrow \text{study}(x, y).$

شكل ٢-٦ : التمثيل الرياضى المنطقى للقواعد

الموجودة فى شكل ٢-٤.

أسس التمثيل الرياضى المنطقى و الحصول على التمثيل الموجود فى

الشكلين ٢-٥ و ٢-٦ و كيفية استخدام ذلك سوف ندرسها بالتفصيل فى الفصل

الرابع بإذن الله تعالى. كذلك الفصل السادس هو تطبيق عملى على استخدام التمثيل

المنطقي لقاعدة معرفة خاصة بالأمراض الإشعاعية مع امكانية الاستعلام باللغة العربية باستخدام لغة PROLOG.

٢-٣ المعرفة القابلة للوراثة Inheritable Knowledge

تمثيل المعرفة القابلة للوراثة أو المعرفة الموروثة و كذا التقنيات التي تعمل هذا النوع من المعرفة مبنية على تقنيات حديثة و هي التقنيات الموجهة الأهداف أو ما يُطلق عليها أحياناً التقنيات الشيئية (Object Oriented Techniques). من أهم تقنيات استنتاج المعرفة هي خاصية الوراثة (Property Inheritance). ننسب الدارسة و الدارس إلى أن تم تخصيص الفصل الثالث للتطبيقات المتعلقة بالمعرفة القابلة للوراثة باستخدام لغة C++.

٢-٣-١ طبيعة المعرفة القابلة للوراثة

Nature of Inheritable Knowledge

المعرفة القابلة للوراثة تعنى أن العناصر أو الكائنات المنتمية إلى صنف أو فئة أو طبقة معينة يمكنها أن تراث خصائص و أنشطة و أحداث من طبقة أو فئة أعم أو أعلى في شجرة تصنيف الكائنات. و لتحقيق ذلك يجب تنظيم و ترتيب الكائنات و الفصائل في شجرة الوراثة. المعرفة القابلة للوراثة مبنية على العناصر التالية :

- كل شيء في الحياة هو كائن (object) ينتمي إلى فئة أو طبقة (class) معينة ينتمي إليها كائنات أخرى تتشابه في مجموعة من الخصائص و الأنشطة. أي أن أي كائن هو عبارة عن مثال أو حالة (instance) من الطبقة التي ينتمي إليها.

- كل فئة أو طبقة يمكن أن يتفرع منها (يُشتق منها) طبقة أو طبقات أخرى و تكون هي الطبقة الأعلى أو الطبقة الأصل أو الطبقة الوالد (parent

(class) للطبقات المتفرعة منها و هي الطبقة الأدنى أو الفرع أو الابن (child class). و من الممكن أن تكون أى طبقة أصل و/أو طبقة فرع فى نفس الوقت، أى تكون مُشتقة من طبقة أب أعلى و يتفرع منها طبقة ابن أو أكثر.

- كل كائن يكتسب كافة خصائص الطبقة التى ينتمى إليها و يرث (inherit) أيضاً كافة خصائص جميع الطبقات التى تعلو الطبقة التى ينتمى إليها هذا الكائن.

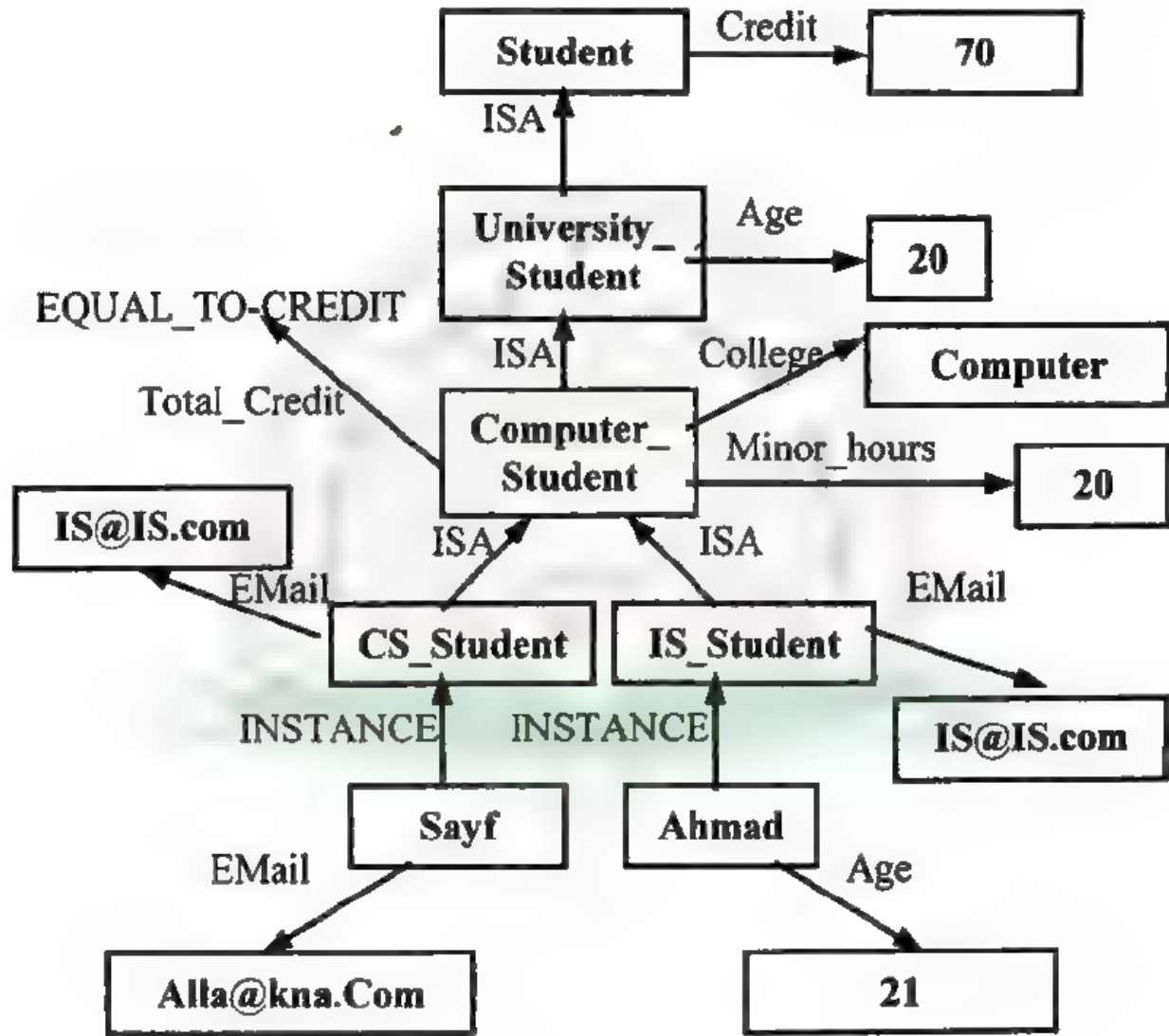
Inheritance Tree

٢-٣-٢ شجرة الوراثة

شجرة الوراثة هى رسم تخطيطى منطقى يمثل علاقة الكائنات بالطبقات التى تنتمى إليها و علاقة الطبقات الابن بالطبقات الأب. و تحتوى على أفرع بين الطبقات و الجذر هو الطبقة الأب الأكبر التى لا تعلوها أى طبقة أخرى مثل طبقة Student (الطالبة/الطالب) فى شكل ٢-٧ الذى يعرض مثالا لشجرة وراثة. كل كائن و كل طبقة له مجموعة من الخصائص تظهر فى الرسم. و نستطيع أن نكتشف أن أى شجرة وراثة تحتوى على عدة عناصر مختلفة و هى :

- الطبقة (class) كما نرى فى شكل ٢-٧ ثلاثة أنواع : النوع الأول هو طبقة جذر (root class) لا يكون لها طبقة أب أعلى و لكنها أصل كل الطبقات التى تتفرع منها و هى أول و أعلى طبقة فى شجرة الوراثة (عكس الشجرة الحقيقية)، و مثال عليها طبقة Student (الطالبة/الطالب). النوع الثانى هو طبقة ورقة (leaf class) و هى طبق متفرعة من طبقة أعلى و لا يتفرع منها أى طبقة أدنى ابن، و مثال عليها الطبقة CS_Student (طالبة/طالب علوم الحاسب) و الطبقة IS_Student (طالبة/طالب نظم المعلومات). النوع الثالث هو الطبقة الوسط بين هذه و تلك فهى

طبقة ابن لطيفة أعلى أب و في نفس الوقت هي طبقة أب يتفرع منها طبقة ابن أو أكثر، مثال عليها الطبقة University_Student (طالبة/طالب الجامعة) و Computer_Student (طالبة/طالب كلية الحاسبات). ترتبط الطبقات مع بعضها البعض بعلاقة رابطة اسمها ISA.



شكل ٢-٧ : شجرة الوراثة.

- الكائن أو المثال أو الحالة (object) و يرتبط مع الطبقة التي ينتمي إليها بعلاقة رابطة اسمها INSTANCE. على سبيل المثال، لدينا الكائن Sayf ينتمي لطبقة طالبة قسم علوم الحاسب CS_Student و الكائن

Ahmad ينتمى لطبقة طلبة قسم نظم المعلومات IS_Student فى شجرة الوراثة الموجودة فى شكل ٧-٢.

- العلاقة الرابطة **ISA relationship** و هى علاقة تربط الطبقة الابن الأدنى بالطبقة الأب الأعلى و يُطلق عليها احتواء الطبقة (**class inclusion**). يتم تمثيل هذه العلاقة فى شجرة الوراثة فى شكل سهم يربط الطبقة الابن بالطبقة الأب، بحيث يبدأ (ذيل) السهم من الطبقة الابن و ينتهى (رأس) السهم عند الطبقة الأب.

- العلاقة الرابطة **INSTANCE relationship** و هى علاقة تربط الكائن بالطبقة التى ينتمى إليها و يُطلق عليها عضوية الطبقة (**class membership**).

- الخصائص (**properties**) و هى صفات أو أنشطة للكائنات أو الطبقات. يتم تمثيلها على الرسم باستخدام سهم يصبه اسم الخاصية و ينتهى إلى مستطيل يحتوى على قيمة الخاصية الأصلية (**default value**) أو القيمة الحقيقية (**real value**).

يمكن تمثيل الطبقة بطرق متعددة منها على سبيل المثال تمثيل فى إطار أو هيكل (**frame**). شكل ٨-٢ يعرض تمثيل طبقة طالبات و طلبة كلية الحاسبات **Computer_Student** فى **frame**.

Computer_Student :

← اسم الإطار و هو اسم الطبقة

ISA : *University_Student*

← مؤشر إلى الطبقة الأب

Minor_Hours : 20

← قيمة ثابتة أولية

College : *Computer*

← قيمة ثابتة أولية

Total_credit : *EQUAL_TO_CREDIT* ← دالة ترجع بقيمة متغيرة

شكل ٢-٨ : تمثيل الطبقة في إطار.

Inheritance algorithm

٢-٣-٣ خوارزم التوريث

نستخدم خوارزم التوريث التالي لاستنتاج قيم من طبقات الشجرة تصف الكائنات المنتمية إلى أحد طبقات الشجرة.

خوارزم التوريث :

To Retrieve a value V for attribute A of an object O :

(لاستخراج قيمة V لخاصية A لكائن O)

1-Find O in the Knowledge base

(اعثر على O في قاعدة المعرفة)

2- IF there is a value for the attribute A, report it .

(إذا كان هناك قيمة الخاصية A اطبعها)

3- Otherwise, See If there is a value for the attribute INSTANCE.

If not , then fail.

(إذا لم يكن، فانظر على قيمة الخاصية INSTANCE، وإذا لم تكن موجودة أنهى بالفشل)

4- Otherwise, move to the node corresponding to that value and look for a value for the attribute A. If found report it.

(إذا لم يكن، تحرك إلى العقدة طبقاً لتلك القيمة، وابحث عن قيمة للخاصية A. إن وجدتھا اطبعھا).

5- Otherwise, Do until there is no value for the ISA attribute or until an answer is found

(إذا لم يكن , استمر إلى ألا يبقى قيمة للخاصية ISA أو حتى العثور على إجابة)

a- Get the value of the IAS attribute and move to that node

(أوجد قيمة ISA وتحرك إلى العقدة التي تشير إليها)

b- see, If there a value for the attribute A, If there is , Report It

(انظر إن كان هناك قيمة للخاصية A اطبعها)

٢-٣-٤ أمثلة محلولة باستخدام خوارزم التوريث

نقدم هنا بعض الأمثلة التي تتطلب استخراج معلومات عن كائنات معينة من شجرة الوراثة باستخدام خوارزم الوراثة. بعض المعلومات نستخرجها مباشرة (مثال ١). البعض الآخر نتحرك على العلاقة الرابطة instance لنأتى بالمعلومة من الطبقة التي ينتمى إليها الكائن (مثال ٢).

أحياناً نتحرك على العلاقة الرابطة isa مرة أو أكثر لنحصل على المعلومة من طبقة أعلى (مثال ٣). و فى بعض الأحيان ننفذ برنامجاً أو دالة للحصول على المعلومة المطلوبة (مثال ٤). و من الممكن بالتأكيد عدم العثور على المعلومة المطلوبة (مثال ٥).

• مثال ١ :

Email (Sayf) ? = Alla@kna.com

موصوف صفة

من الخطوات ١ و ٢ عن طريق الكائن Sayf و خاصيته Email.

• مثال ٢ :

Email (Ahmad) = IS@IS.com

من الخطوات ١ و ٢ و ٣ و ٤ عن طريق Instance من الكائن Ahmad إلى طبقة طالبات و طبقة نظم المعلومات IS_Student و خاصيتها EMail. لاحظ أنه لا توجد قيمة لخاصية البريد الإلكتروني Email للكائن Ahmad و لذلك اعتبرنا قيمة البريد الإلكتروني للقسم الذي يدرس به.

• مثال ٢ :

College (Sayf) ? = Computer

من الخطوات ١-٥ عن طريق Instance من الكائن Sayf إلى طبقة طالبات و طبقة علوم الحاسب CS_Student إلى طبقة طالبات و طبقة كلية الحاسبات Football_Player عن طريق ISA.

• مثال ٤ :

Credit (Ahmad) ? = 70

من الخطوة ١-٥ عن طريق Instance من الكائن Ahmad إلى الطبقة IS_Student و عن طريق ISA إلى الطبقة Computer_Stuent ثم تنفيذ الدالة EQUAL_TO_HANDED التي تأتى بقيمة Credit خاصية الطبقة Student.

• مثال ٥ :

Speed (Sayf) ? = unknown

بتطبيق جميع خطوات الخوارزم نصل إلى آخر خطوة و إلى طبقة الجذر و لا نجد الخاصية أبداً فتكون الإجابة غير معلومة unknown.

٢-٤ شبكة المعرفة اللفظية والإطار

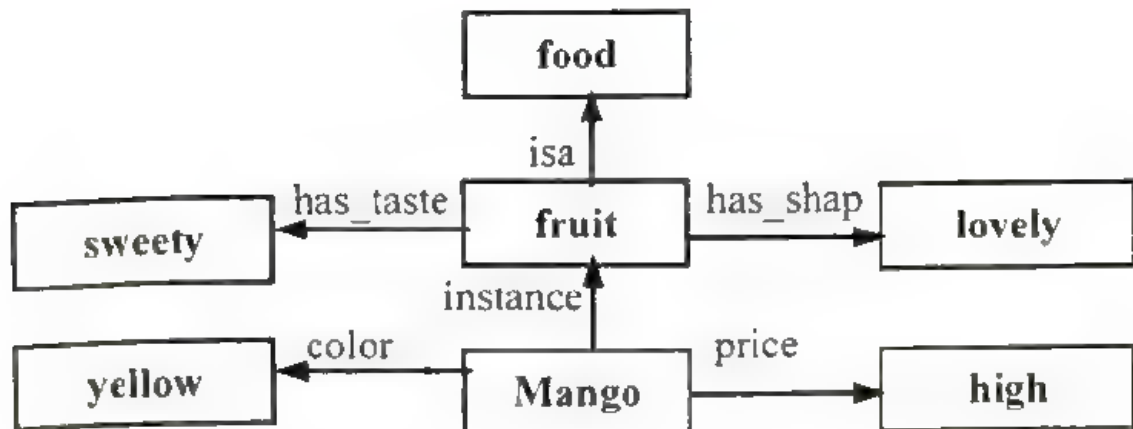
Semantic Net and Frame

نُعرّف هنا على نوعين للتراكيب المستخدمة في تمثيل المعرفة و التقنيات المستخدمة في الاستدلال على الحلول معهم. النوعان هما شبكة المعرفة اللفظية و الإطار. الجدير بالذكر أن في الكثير من الأحيان يُستخدم النوعين معاً في نفس قاعدة المعرفة.

٢-٤-١ شبكة المعرفة اللفظية Semantic Net

في اللغات الطبيعية مثل اللغة العربية أو الإنجليزية أو أى لغة أخرى، في كثير من الأحيان، لا نفهم معنى الكلمات بمفردها و لكن نفهمها من خلال جمل تضم تلك الكلمات. كذلك فإن الفكرة الرئيسية وراء شبكة المعرفة اللفظية تكمن في أن معنى المفهوم يأتي من طرق ربطه أو اتصاله بالمفاهيم الأخرى.

يتم تمثيل المعرفة في شبكة المعرفة اللفظية كمجموعة من العقد أو النقاط (nodes) متصلة ببعضها البعض بواسطة مجموعة من الأقواس أو الأسهم التي تحمل عنواناً لتمثل العلاقات الرابطة بين العقد. شكل ٢-٩ يعرض مثالاً لشبكة معرفة لفظية.



شكل ٢-٩ : شبكة معرفة لفظية لطبقات منتجات استهلاكية.

الشبكة الموجودة فى شكل ٢-٩ تتضمن طبقة الفاكهة fruit ترتبط بطبقة أعلى هى طبقة الطعام food من خلال العلاقة الرابطة isa. تحتوى الشبكة أيضاً على كائن المانجو mango المنتمى إلى طبقة الفاكهة عن طريق العلاقة الرابطة instance.

نرى أيضاً الخاصيتين has_taste (له مذاق) لطبقة الفاكهة و قيمتها sweet (حلو المذاق) و has_shape (له شكل) و قيمتها lovely (فاتنة). و الخاصيتين color (اللون) و قيمتها yellow (أصفر) و price (السعر) و قيمتها expensive (غالية). نستطيع استخدام خاصية الوراثة لنحصل على معلومة تفيد أن المانجو حلوة المذاق و فاتنة الشكل أى :

has_taste (mango, sweet).

has_shape (mango, lovely).

يمكننا الحصول على تمثيل منطقى مماثل للخاصيتين اللتين تصفان الكائن mango و هما price و color كما يلى :

color (mango, yellow).

price (mango, expensive).

شبكة المعرفة اللفظية طريقة طبيعية لتمثيل العلاقات الرابطة التى يمكن تمثيلها باستخدام التمثيل الرياضى المنطقى كما سنرى فى الفصل الرابع.

Frame

٢-٤-٢ الإطار

الإطار أو الهيكل (frame) هو مجموعة الخصائص (تسمى أحياناً فتحات أو فراغات slots) و القيم المصاحبة (و ممكن القيود على القيم) التى تصف كياناً

معيناً. فى الغالب لا يُستخدَم إطاراً بمفرده و لكن يُستخدَم نظام إطار (frame system) يتكون من مجموعة من الإطارات مرتبطة ببعضها البعض. تربط الإطارات ببعضها البعض عن طريق خاصية تكون قيمتها اسماً لإطار آخر.

نستطيع استخدام نظام الإطار لتمثيل طبقة (class) أو كائن (object) و خصائصهما و العلاقات الرابطة بينهم. تأمل الشكل ٢-١٠ الذى يعرض نظام إطار لتمثيل شبكة المعرفة اللفظية الموجودة فى شكل ٢-٩.

fruit :

isa : food
has_tasty : sweet
has_shape: lovely

mango :

instance : fruit
color : yellow
price : expensive

شكل ٢-١٠ : نظام إطار لتمثيل شبكة المعرفة اللفظية
الموجودة فى شكل ٢-٩.

٢-٥ المعرفة الإجرائية Procedural Knowledge

المعرفة التى تمثيلها فى شجرة الوراثة الخاصة بطالبات و طلبية الحاسبات و المعلومات فى الجزء السابق تركز على حقائق ثابتة و خبرية. يوجد نوع آخر من المعرفة على درجة كبيرة من الأهمية هى المعرفة الإجرائية. يتم تمثيل المعرفة الإجرائية من خلال الإجراءات -دوال أو برامج صغيرة- (Procedures).

هذه المعرفة يمكن تمثيلها بعدة طرق مثل البرمجة، لكنه من الصعوبة بمكان كتابة برنامج ليستنتج أو يستدل على الحل من برنامج آخر. أيضاً، عملية

كتابة و تعديل البرامج الكبيرة صعب جداً. لذلك يتم تمثيل المعرفة الإجرائية في صورة يسهل تناولها من قبل البرامج و المستخدمين. أشهر هذه الطرق هي قواعد الإنتاج (Production rules). حيث تأخذ هذه القواعد شكل الجمل الشرطية مثل الموجودة في شكل ١١-٢.

If : A student finish 20 Minor credit_houres, and
finished 50 Major credit_hours, and
he is CS_Student
Then He is a graduated.

شكل ١١-٢ : مثال للمعرفة الإجرائية كقواعد شرطية.

المثال الموجود في شكل ١١-٢ يبين أن الطالب الذي ينهى ٢٠ ساعة معتمدة في المقررات العامة (minor) و ينهى ٥٠ ساعة معتمدة في مقررات التخصص (Major) و يدرس بقسم علوم الحاسب الآلى يصبح خريجاً.

٢-٦ التمثيل باستخدام الحالات Case_Based Knowledge

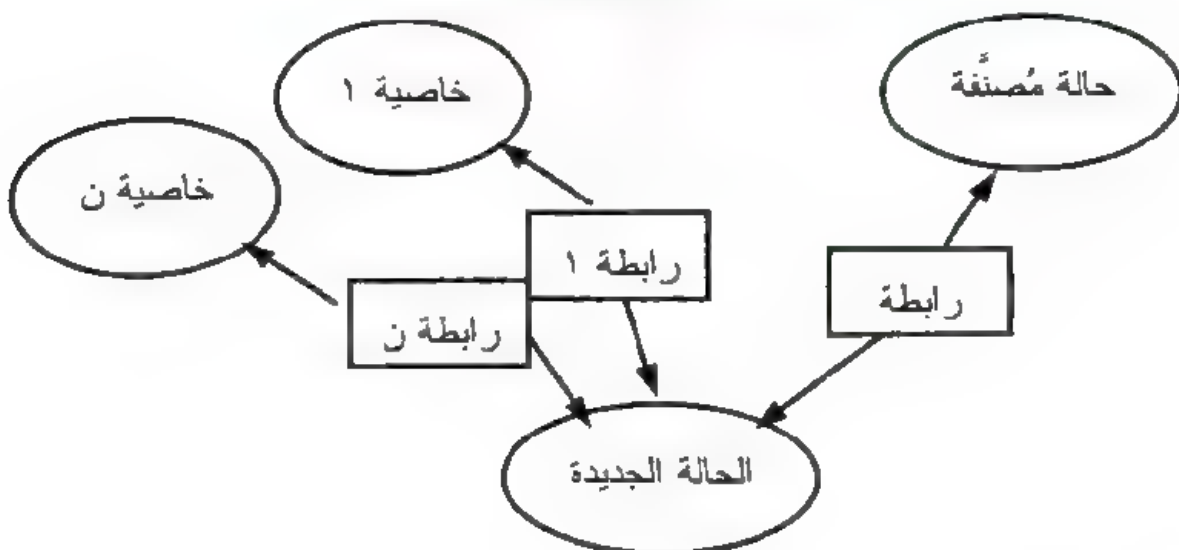
خبرة الخبراء في مجال ما تتكون من العديد من الحالات (cases) أو الأمثلة أو التجارب الفعلية العملية بالإضافة إلى العناصر الأخرى من خبرتهم العملية. عملية تذكّر و تقييم الحالات المستقلة أسهل بكثير من تطوير قواعد عمومية. في تلك الحالات، يعرف الخبراء أكثر من قدرتهم على التعبير. في الكثير من المواقف يستطيع الخبراء المتميزين أن يتدبروا أمورهم بشكل جيد مع مشاكل القرارات الصعبة، لكن عندما يُطلب منهم تفسير ما يفعلوا فإنهم يواجهون صعوبة.

النظم المبنية على الحالات (case_based systems) تعالج مشكلة التمثيل من خلال إدخال أمثلة مستقلة لحالات معينة. ميزة هذه الطريقة هي أن تحليل و تشخيص مواقف (أمثلة) معينة أكثر طبيعية للخبراء و لا تتطلب منهم مجهوداً

إضافياً. شكل ١٢-٢ يعرض مثلاً لوصف حالة فعلية لمريض في نظام ناصر-٩٦ المبني على تمثيل الحالات و يعرض شكل ١٣-٢ رسماً تخطيطياً لتمثيل الحالة الجديدة و ارتباطها بخصائصها و بحالة قديمة مُصنَّفة من قبل (سندرس هذا النظام بالتفصيل في الفصل الثامن)

<u>Case Identification Data</u> (بيانات تعريف الحالة)			
Case Id. : Sayfo Allah		Area : Cairo	
Date : 1/10/1996		Phone : 747120	
<u>Case Features</u> (خصائص الحالة)			
FNo.	Feature Name اسم الخاصية	Value1 القيمة ١	Value2 القيمة ٢
F1	wheezing		
F2	cough	early morning	night
F3	dyspnoea		
F4	night symptoms	> 2/month	
F5	Peak Expiration Flow	>=80%	
...			
Fn			

شكل ١٢-٢ : حالة لمريض تبين وصف الحالة.



شكل ١٣-٢ : الحالة الجديدة و روابطها بخصائصها و حالة مُصنَّفة.

تواجه النظم المبنية على الحالات مشكلتين. المشكلة الأولى تتعلق بتطوير قاعدة معرفة تتكون من مجموعة تمثيلية من الحالات، أى تعبر عن المجال و تمثله بشكل جيد. المشكلة الثانية تتعلق بفهرسة الحالات (indexing of cases) و البحث عن طريق سريعة للبحث فى القرارات السابقة للوصول إلى الحل.

إذا وُجِدَت مجموعة كبيرة من الحالات تمثل مواقف بالكامل فمن المعقول استخدام وصف للحالات من بعض قواعد البيانات. فى العادة، يكون ضرورياً أن نستخدم الخبراء كمصادر لوصف الحالات. عملية مقابلة خبير للاستشارة فى قاعدة المعرفة يمكن تنفيذها مع استخدام بعض تقنيات استخلاص البيانات.

هناك تطبيق فى الفصل الثامن على استخدام التمثيل بالحالات و الإطار و شبكة المعرفة اللفظية و الاستدلال على الحلول بالحالات و تصنيفها بلغة C++.

٧-٢ أسئلة

١. عرف Knowledge Representation و أهميتها لبرامج AI.
٢. ما المقصود بالعلاقتين isa و instance؟ مع مثال لكل منهما.
٣. وضح الفرق بين internal representation و external representation للحقائق.
٤. وضح الفرق بين الحقائق و القواعد فى طريقة التمثيل الرياضى المنطقى مع مثال لكل منهما و تمثله.
٥. وضح طبيعة المعرفة القابضة للوراثة و الأسس المبنية عليها.

٦. عرّف الطبقة و الكائن مع مثال لكل منهما.

٧. استخدم خوارزم التوريث الموجود مع شجرة الوراثة الموجودة في شكل ٢
٧- للإجابة على الأسئلة التالية :

أ- Credit(Sayf).

ب- Minor_credit(Ahmad).

ت- College(Ahmad).

٨. ما المقصود بشبكة المعرفة اللفظية مع مثال.

٩. وضّح المقصود بالإطار و نظام الإطار مع مثال لشبكة معرفة و تمثيلها في نظام إطار.

١٠. عرّف المعرفة الإجرائية مع مثال.

١١. ما المقصود باستخدام الحالات في تمثيل المعرفة و مثال لحالة.

الفصل الثالث

المعرفة الموروثة و شبكة المعرفة اللفظية بلغة C++

Inheritable Knowledge and Semantic Net Using C++

توفّر لغة C++ إمكانيّتين هامتين في مجل البرمجة موجهة الأهداف (Object-Oriented Programming). الإمكانيّة الأولى هي الوراثة و الإمكانيّة الثانية هي الدوال التخليّة. و تُعدّ الوراثة التقنيّة الرئيسيّة لوصف و تناول العلاقات الرابطة بين الطبقات. إمكانيّة الوراثة (Inheritance) هي القدرة على اشتقاق فئة/طبقة (class) أو أكثر من فئة/طبقة أخرى أو أكثر بحيث ترث طبقة أعضاء طبقة أخرى من بيانات و دوال بالإضافة إلى البيانات و الدوال التي تخصها. و تُعدّ الوراثة أحد طرق تمثيل المعرفة في تطبيقات الذكاء الاصطناعي.

Inheritance

١-٣ الوراثة

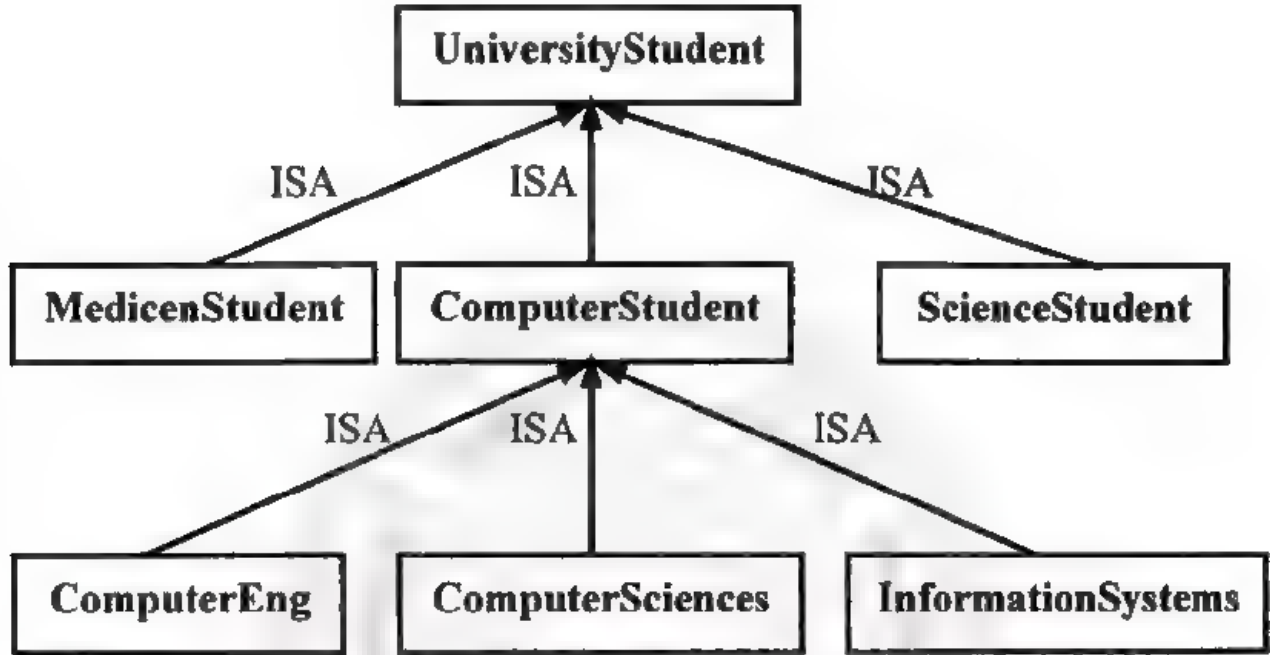
من أهم مميزات لغة ++C هي الوراثة أو الاشتقاق. فهذه الميزة تسمح للطبقات بأن تَرث أداء طبقات أخرى و أن تعدّل ذلك الأداء. تُسمى الطبقة التي يُشتق منها بالطبقة القاعدة (Base Class) أو الطبقة الأعلى أو طبقة الأب (father class).

عند اشتقاق طبقة من طبقة أخرى تَرث الطبقة المُشتقة خصائص الطبقة القاعدة. فنَرث الطبقة المُشتقة كافة أعضاء الطبقة القاعدة من متغيرات (عامة public و خاصة private محمية protected) و من دوال عامة. الجدير بالذكر أن الطبقة المُشتقة تَرث فعلياً المتغيرات الخاصة private و لكنها لا تستطيع تناولها أو التعامل معها.

و بالإضافة إلى ما تَرثه الطبقة المُشتقة من الطبقة القاعدة، فإن الطبقة المُشتقة لها المتغيرات و الدوال الأعضاء بها (عامة و خاصة و محمية). يمكن أن نشق طبقة أخرى أو أكثر من الطبقة القاعدة. كذلك يمكن أن نشق طبقة أو أكثر من الطبقة المُشتقة في تسلسل هرمي لبناء شجرة الوراثة مثل التي نراها في الشكل ١-٣.

الشكل ١-٣ يبين لنا الطبقة القاعدة UniversityStudent (طالب الجامعة) تتفرع منها ثلاث طبقات هي طبقة MedicineStudent (طالب الطب) و طبقة ComputerStudent (طالب الحاسب) و طبقة ScienceStudent (طالب العلوم). بعد ذلك نشق من طبقة طالب الحاسب ثلاث طبقات هي ComputerEng (طالب هندسة الحاسب) و ComputerSciences (طالب علوم الحاسب) و طبقة InformationSystems (طالب نظم المعلومات).

تعتبر الطبقة الأولى القاعدة هي جذر الشجرة و تتفرع منها طبقات أخرى و كل طبقة يمكن أن يتفرع من طبقة أو طبقات أخرى إلى أن نصل إلى الطبقات التي تمثل الأوراق.



شكل ٣-١ : شجرة طبقات طلبة الجامعة و اشتقاقها من بعضها البعض.

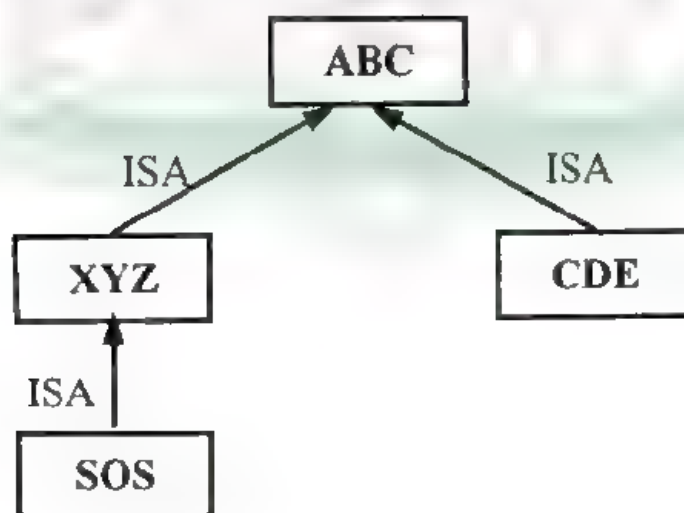
الجدير بالذكر أن الطبقة المشتقة من طبقة هي بدورها مشتقة من طبقة أعلى و هكذا حتى نصل إلى الطبقة الجذر، ترث كافة أعضاء الطبقات (متغيرات و دوال) التي تعلوها بالإضافة إلى أعضائها المنتمين إليها.

في حالة إعادة تعريف عضو موروث (متغير أو دالة) في الطبقة المشتقة، فإن الاسم المعاد تعريفه (الموجود في الطبقة المشتقة) يغطي و يخفي الاسم الموروث في إطار الطبقة المشتقة فقط. لكي نتناول اسم متغير مخفي (في الطبقة القاعدة) لابد من ذكر اسم طبقة القاعدة التي يتبعها الاسم المخفي ثم المعامل ":" ثم اسم المتغير المخفي.

عملية الوراثة تتضمن فائدتين فى برامج C++. الفائدة الأولى هى أن الطبقات المشتقة توفر وسيلة لبناء تسلسل هرمى للطبقات التى تمثل واقعاً فى حياتنا. الفائدة الثانية هى إمكانية اشتقاق طبقات متشابهة و لكن غير متطابقة. يجرى ذلك فى سياق بناء شبكة معرفة لفظية (semantic net) لمجال المعرفة فى تطبيق معين.

٢-٣ الوراثة المفردة Single Inheritance

أغلب حالات الوراثة تقع تحت تصنيف الوراثة المفردة. نعى بالوراثة المفردة أن نشق طبقة من طبقة واحدة أعلى فى الشجرة مثل جميع الطبقات المشتقة الموجودة فى الشكلين ١-٣ و ٢-٣. حيث يوجد لكل طبقة (طبقة ابن) طبقة واحدة أعلى (طبقة أب)، بينما يمكن أن نشق أى عدد من الطبقات الأبناء من الطبقة الأعلى الأب.



شكل ٢-٣ : مثال عام على شجرة الوراثة المفردة.

الآن دعنا ننظر إلى شجرة الوراثة الموجودة فى الشكل ٢-٣. تحتوى شجرة الوراثة على طبقة الجذر ABC و قد اشتققنا طبقتين منها هما XYZ و CDE. ثم اشتققنا الطبقة SOS من الطبقة XYZ. لاحظ أن كلاً من الطبقتين

XYZ و CDE قد ورثا أعضاء الطبقة ABC إلى جانب أعضاء كل طبقة. أما الطبقة SOS فقد ورثت أعضاء الطبقتين XYZ و ABC إلى جانب الأعضاء المنتمين إليها. الطريقة العامة لتعريف طبقات تلك الشجرة كما يلي :

```
class ABC
{
    ..
    ..
};
class XYZ:public ABC
{
    ..
    ..
};
class CDE:public ABC
{
    ..
    ..
};
class SOS:public XYZ
{
    ..
    ..
}
```

مثال ١ : مطلوب بناء طبقة لمركبات النقل vehicle بنوعها سيارات الركوب الصغيرة car و الحافلات العامة bus. حيث يخص طبقة مركبات النقل المتغيرات plate_no (رقم اللوحة) و model (الموديل). بينما يخص طبقة سيارات الركوب المتغير doors (عدد الأبواب) و يخص طبقة الحافلات المتغير chairs (عدد المقاعد). للتسهيل اعتبر جميع المتغيرات من النوع العام public و أن اشتقاق

الكائنات من تلك الطبقات يتم باستخدام معامل النقط. اكتب البرنامج الذي يعرف تلك الطبقات و يشتق كائنات مفردة و مصفوفة كائنات من كل طبقة و يعالج بيانات كلاً منهم.

الحل :

البرنامج الموجود في الشكل ٣-٣ يُعرف الطبقات الثلاثة. نرى في البرنامج دالة البناء للطبقة الأولى القاعدة vehicle و قد تناولت المتغيرين الأعضاء في تلك الطبقة و هما plate_no و model.

نرى أيضاً طبقة البناء للطبقة المشتقة car و تضم ثلاث متغيرات هم plate_no و model و doors لكنها تعالج المتغير الخاص بطبقتها فقط و هو doors بينما تمرر المتغيرين الأوليين plate_no و model إلى دالة البناء في الطبقة الأعلى vehicle.

بالمثل تفعل دالة البناء للطبقة المشتقة bus مع متغيراتها الثلاثة plate_no و model و chairs، حيث تعالج متغير طبقتها chairs فقط و تمرر المتغيرين الأوليين plate_no و model العضوين في الطبقة القاعدة vehicle.

```
#include<iostream.h>
#include<string.h>
// طبقة مركبة النقل
class vehicle{
public:
    vehicle ( ) { }
    vehicle (char p[7], int m);
    char plate_no[7];
    int model;
};
```

// دالة البناء لكائنات طبقة مركبة النقل

```
vehicle::vehicle(char p[7], int m)
```

```
{
    strcpy(plate_no, p);
    model=m;
}
```

طبقة سيارة الركوب الصغيرة المشتقة من طبقة مركبة النقل

```
class car:public vehicle{
```

```
public :
```

```
    car ( ) {}
```

// المتغيرين الأولين موروثن من مركبة النقل و الثالث خاص بطبقة سيارة الركوب

```
    car(char p[7], int m, int d);
```

```
    int doors;
```

```
};
```

// دالة البناء لسيارة الركوب تستدعي دالة البناء لطبقة مركبة النقل و تمرر لها *

*/المتغيرين الأعضاء بها و تعالج المتغير الثالث العضو بطبقة سيارة الركوب.

```
car::car(char p[7], int m, int d):vehicle(p,m)
```

```
{
    doors = d;
```

```
}
```

طبقة الحافلة المشتقة من طبقة مركبة النقل

```
class bus:public vehicle{
```

```
public:
```

```
    bus ( ) {}
```

```
    bus(char p[7],int m,int c);
```

```
    int chairs;
```

```
};
```

// دالة البناء للحافلة تستدعي دالة البناء لطبقة مركبة النقل و تمرر لها المتغيرين *

*/الأعضاء بها و تعالج المتغير الثالث العضو بطبقة الحافلة.

```
bus::bus(char p[7],int m,int c):vehicle(p,m)
```

```
{
    chairs=c;
```


}

//الدالة الرئيسية

int main ()

{

vehicle V1,V2("ABC 121",1999),V3[10];

car C1,C2("XYZ 321",2001,5),C3[10];

bus B1,B2("KKK 999",2003,4),B3[10];

//إدخال بيانات كائن مركبة نقل

cout <<"\nEnter plate_no,model\n";

cin >> V1.plate_no >> V1.model;

//إدخال بيانات كائن سيارة ركوب

cout <<"\nEnter plate_no,model,doors\n";

cin >> C1.plate_no >> C1.model >> C1.doors;

//إدخال بيانات كائن حافلة

cout <<"\nEnter plate_no,Model,Chairs";

cin >> B1.plate_no >> B1.model >> B1.chairs;

//طباعة بيانات كائن مركبة نقل

cout <<"\n plate_no : " << V1.plate_no;

cout <<"\n Model : " << V1.model;

//طباعة بيانات كائن سيارة ركوب

cout <<"\n plate_no : " << C1.plate_no;

cout <<"\n Model : " << C1.model;

cout <<"\n Doors : " << C1.doors;

//طباعة بيانات كائن حافلة

cout <<"\n Plate_no : " << B1.plate_no;

cout <<"\n Model : " << B1.model;

cout <<"\n chairs : " << B1.chairs;

//طباعة بيانات كائن حافلة

cout <<"\n Plate_no : " << B2.plate_no;

cout <<"\n Model : " << B2.model;

cout <<"\n Chairs : " << B2.chairs;

cout <<"\n";

int i,n;

```

// معالجة بيانات مصفوفة كائنات لطبقة سيارة الركوب
cout << "\n Enter Number Of Cars : ";
cin >> n;
for(i=0;i<n;i++)
{
    // إدخال و طباعة بيانات كائن في مصفوفة كائنات سيارة الركوب
    cout << "\n Enter plate_no,Model,doors";
    cin >> C3[i].plate_no >> C3[i].model >> C3[i].doors;
    cout << "\n Plate_no : " << C3[i].plate_no;
    cout << "\n Model : " << C3[i].model;
    cout << "\n Doors : " << C3[i].doors;
};

// معالجة بيانات مصفوفة كائنات لطبقة الحافلة
cout << "\n Enter Number Of bus : ";
cin >> n;
for(i=0;i<n;i++)
{
    // إدخال و طباعة بيانات كائن في مصفوفة كائنات الحافلات
    cout << "\n Enter plate_no,Model,chairs";
    cin >> B3[i].plate_no >> B3[i].model >> B3[i].chairs;
    cout << "\n Plate_no : " << B3[i].plate_no;
    cout << "\n Model : " << B3[i].model;
    cout << "\n Chairs : " << B3[i].chairs;
};

// معالجة بيانات مصفوفة كائنات لطبقة مركبة النقل
cout << "\n Enter Number Of Vhicle : ";
cin >> n;
for(i=0;i<n;i++)
{
    // إدخال و طباعة بيانات كائن في مصفوفة كائنات مركبة النقل
    cout << "\n Enter plate_no,Model";
    cin >> V3[i].plate_no >> V3[i].model;

```

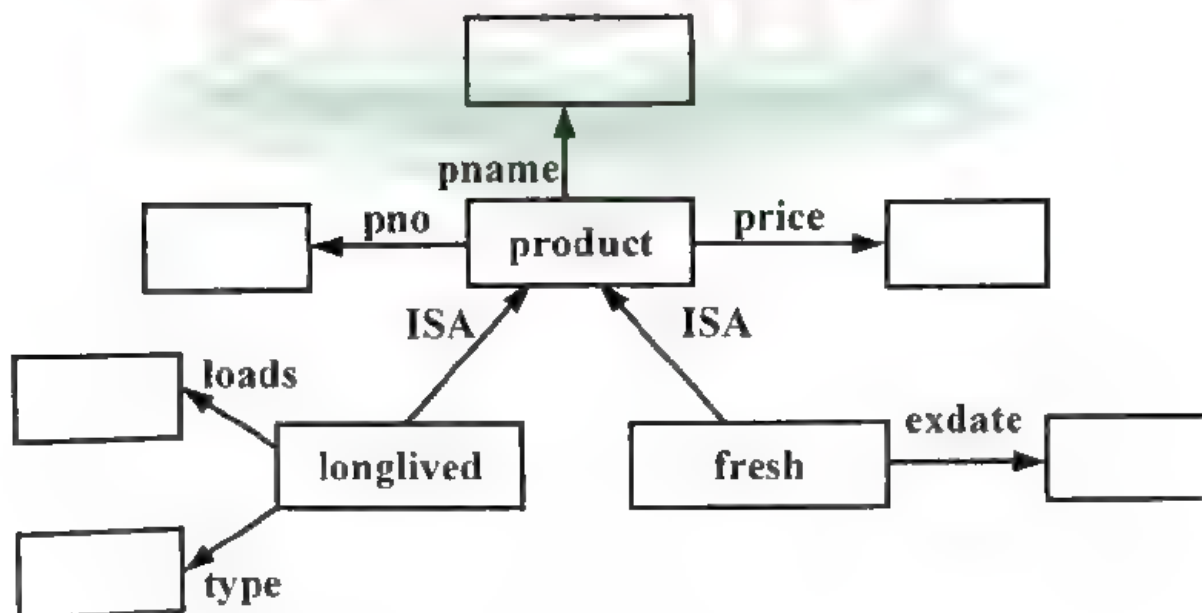
```

    cout << "\n Plate_no : " << V3[i].plate_no;
    cout << "\n Model : " << V3[i].model;
};
cout << "\n";
return 0;
}

```

شكل ٣-٣ : برنامج معالجة الوراثة بين طبقات مركبات النقل.

مثال ٢ : المطلوب في هذا المثال هو تعريف طبقات المنتجات الاستهلاكية الثلاث الموجودة في الشكل ٣-٤ في برنامج يعالج بيانات الطبقة القاعدة product (المنتج) و الطبقتين المشتقتين منها fresh (المنتج الطازج) و longlived (المنتج المعمّر). مع تعريف جميع المتغيرات من النوع private. بين كيفية نسخ كائنات بين الطبقات الثلاث.



شكل ٣-٤ : شجرة طبقات المنتجات الاستهلاكية.

الحل :

البرنامج الموجود في الشكل ٣-٥ يعالج طبقات المنتجات الاستهلاكية الموجودة في الشكل ٣-٤. حيث نقوم بتعريف كل طبقة بما تحتويه من أعضاء كمتغيرات خاصة. فنرى الطبقة القاعدة product و قد احتوت على المتغيرات الخاصة pno (رقم المنتج) و pname (اسم المنتج) و price (سعر المنتج). كذلك تضم الطبقة product دوال معالجة متغيرات الطبقة.

نرى أيضاً أن الطبقة المشتقة longlived قد ورثت المتغيرات الثالثة الأعضاء بالطبقة القاعدة product مع دوال معالجتها إلى جانب المتغيرين الخاصين العضوين بها loads (عدد الوحدات الممكن رصها فوق بعضها) و type (نوع المنتج المعمر) و دوال معالجة هذين المتغيرين.

كذلك نرى الطبقة المشتقة fresh وقد ورثت المتغيرات الثالثة الأعضاء بالطبقة القاعدة product مع دوال معالجتها إلى جانب المتغير الخاص بها exdate (تاريخ انتهاء الصلاحية) و دوال معالجة هذا المتغير.

لاحظ أن دالة بناء أى طبقة مشتقة تتلقى قيم متغيرات الطبقة المشتقة و المتغيرات الموروثة. ثم تقوم بتمرير قيم المتغيرات الموروثة إلى دالة بناء الطبقة القاعدة قبل أن تعالج متغيرات الطبقة التى تنتمى إليها.

لاحظ كيفية نسخ بيانات كائن من طبقة معينة إلى كائن من نفس الطبقة. بعد ذلك لاحظ كيفية نسخ بيانات كائن من طبقة مشتقة إلى كائن من طبقة أعلى (قاعدة). أخيراً، لاحظ كيفية نسخ بيانات كائن من طبقة أعلى إلى كائن من طبقة مشتقة.

```

#include<iostream.h>
#include<string.h>
// طبقة المنتج الاستهلاكي
class product{
public :
    product () {}
    product (int n, char *pn, float p);
    void setpno(int n) { pno = n; }
    int getpno() { return pno; }
    void setprice(float p) { price=p; }
    float getprice() { return price; }
    void setpname(char *pn) { strcpy (pname,pn); }
    char *getpname() { return pname; }
private :
    int pno;
    float price;
    char pname[40];
};
// دالة بناء طبقة المنتج الاستهلاكي
product::product(int n,char *pn,float p)
{
    setpno(n);           //pno=n;
    setpname(pn);        //strcpy(pname,pn);
    setprice(p);         //price=p;
}
// طبقة المنتج المعمر مشتقة من طبقة المنتج الاستهلاكي
class longlived:public product{
public:
    longlived( ) { }
    longlived(int n,char *pn,float p,int lon,char *t);
    void setloads(int lon) { loads=lon; }
    int getloads( ) { return loads; }
    void settype(char *t) { strcpy(type,t); }
    char *gettype( ) { return type; }

```


private :

```
int loads;
char type[40];
```

```
};
```

// دالة بناء المنتج المعمر تمرر المتغيرات الثلاث الأولى إلى دالة بناء طبقة المنتج

```
longlived::longlived(int n,char *pn,float p,int lon,char *t):
```

```
product(n, pn, p)
```

```
{
```

```
    setloads(lon);    //loads=lon;
```

```
    settype(t);       //strcpy(type,t);
```

```
}
```

// طبقة المنتج الطازج مشتقة من طبقة المنتج اللاستهلكي

```
class fresh:public product
```

```
{
```

public :

```
    fresh ( ) { }
```

```
    fresh (int n,char *pn,float p,char *ed);
```

```
    void setexdate(char *ed) { strcpy(exdate,ed); }
```

```
    char *getexdate() { return exdate; }
```

private :

```
    char exdate[10];
```

```
};
```

// دالة بناء طبقة المنتج الطازج تمرر المتغيرات الثلاث الأولى إلى طبقة المنتج

```
fresh::fresh(int n,char *pn,float p,char *ed):product(n,pn,p)
```

```
{
```

```
    setexdate(ed);
```

```
}
```

```
int main( )
```

```
{
```

```
    int A,B;
```

```
    float C;
```

```
    char name[40],type[40],data[40];
```

```
product p1,p2(5,"suger",5),p3[5];
longlived L1,L2(7,"TV",1000,5,"Electronics"),l3[5];
fresh F1,F2(9,"Milk",6,"5 Ramadan 24"),F3[5],*Fptr;
```

// طباعة بيانات كائن من طبقة المنتج الاستهلاكي

```
cout << "\n Number is : " << p2.getpno();
cout << "\n Name is : " << p2.getpname();
cout << "\n Price is : " << p2.getprice();
```

// طباعة بيانات كائن من طبقة المنتج المعمر

```
cout << "\n Number : " << L2.getpno();
cout << "\n Name : " << L2.getpname();
cout << "\n Price : " << L2.getprice();
cout << "\n Load Number : " << L2.getloads();
cout << "\n Type : " << L2.gettype();
```

// طباعة بيانات كائن من طبقة المنتج الطازج

```
cout << "\n Number : " << F2.getpno();
cout << "\n Name : " << F2.getpname();
cout << "\n Price : " << F2.getprice();
cout << "\n Expire date : " << F2.getexdate();
```

// إدخال بيانات كائن من طبقة المنتج الاستهلاكي

```
cout << "please enter number & name & price product :";
cout << "\n Product No. : "; cin >> A;
cout << "\n Name : "; cin >> name;
cout << "\n Price : "; cin >> C;
p1.setpno(A);
p1.setprice(C);
p1.setpname(name);
```

// طباعة بيانات كائن من طبقة المنتج الاستهلاكي

```
cout << "\n Number : " << p1.getpno();
cout << "\n Name : " << p1.getpname();
cout << "\n Price : " << p1.getprice();
```

// إدخال بيانات كائن من طبقة المنتج المعمر

```
cout << "\nNumber, name, price, loads, type Longlived :";
cout << "\n Product No. : ";      cin >> A;
cout << "\n Name : ";             cin >> name;
cout << "\n Price : ";            cin >> C;
cout << "\n Loading number : ";   cin >> B;
cout << "\n Type : ";             cin >> type;
L1.setpno(A);
L1.setprice(C);
L1.setpname(name);
L1.setloads(B);
L1.settype(type);
```

// طباعة بيانات كائن من طبقة المنتج المعمر

```
cout << "\n Number : " << L1.getpno();
cout << "\n Name : " << L1.getpname();
cout << "\n Price : " << L1.getprice();
cout << "\n Load Number : " << L1.getloads();
cout << "\n Type : " << L1.gettype();
```

// إدخال بيانات كائن من طبقة المنتج الطازج

```
cout << "\nNumber, name, price, expire date Fresh :";
cout << "\n Product No. : ";      cin >> A;
cout << "\n Name : ";             cin >> name;
cout << "\n Price : ";            cin >> C;
cout << "\n Expire date : ";      cin >> data;
F1.setpno(A);
F1.setprice(C);
F1.setpname(name);
F1.setexdate(data);
// طباعة بيانات كائن من طبقة المنتج الطازج
cout << "\n Number : " << F1.getpno();
cout << "\n Name : " << F1.getpname();
cout << "\n Price : " << F1.getprice();
cout << "\n Expire date : " << F1.getexdate();
```

```

طباعة بيانات كائن باستخدام المؤشر //
Fptr = &F1;
cout << "\n Number : " << Fptr->getpno();
cout << "\n Name   : " << Fptr->getpname();
cout << "\n Price   : " << Fptr->getprice();
cout << "\n Expire date : " << Fptr->getexdate();
// نسخ بيانات كائن إلى كائن من نفس الطبقة
p1 = p2;
cout << "\n Number : " << p1.getpno();
cout << "\n Name   : " << p1.getpname();
cout << "\n Price   : " << p1.getprice();
// نسخ بيانات كائن إلى كائن من طبقة أعلى
p1=F2;
cout << "\n Number : " << p1.getpno();
cout << "\n Name   : " << p1.getpname();
cout << "\n Price : " << p1.getprice();

cout << "\n\n\n";
return 0;

}

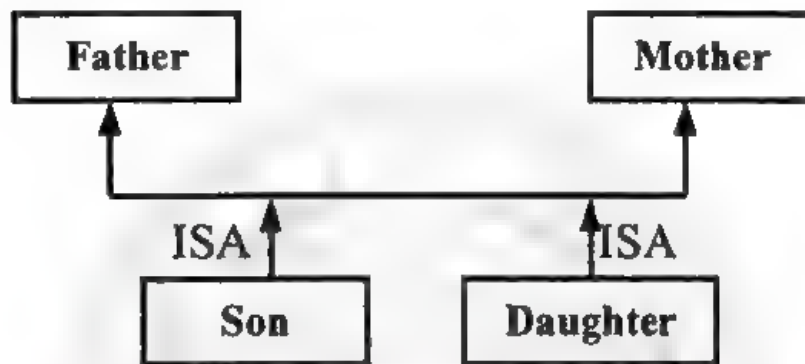
```

شكل ٣-٥ : برنامج معالجة الوراثة بين طبقات المنتجات الاستهلاكية.

٣-٣ الوراثة المتعددة Multiple Inheritance

اشتقاق طبقة أو مجموعة طبقات معينة من طبقة واحدة غلبَ على ما درسنا فيما سبق و أطلقنا عليه الوراثة المفردة لأن الطبقة القاعدة كانت واحدة فقط. لكن في كثير من الأحيان نشق طبقة أو أكثر من طبقتين أو أكثر في نفس الوقت، أي أن هناك أكثر من طبقة قاعدة. لذلك فإن الطبقة أو الطبقات المشتقة ترث خصائص (متغيرات و أداء) جميع الطبقات الأعلى و يُطلق على هذه الحالة الوراثة المتعددة لتعدد الطبقة القاعدة.

أقرب مثال فى الحياة على الوراثة المتعددة هو خلق الله سبحانه و تعالى للإنسان. فكما نعلم يرث الجنين بعض الصفات الوراثية من الأم و البعض الآخر من الأب بالإضافة إلى بعض الصفات الخاصة به ليتشكل لنا شخص جديد يختلف فى كثير من الصفات حتى عن أبويه و أخواته. أيضاً بعد أن يولد الإنسان يكتسب صفات و طباع ممن حوله و أولهم الأب و الأم و الأخوة. شجرة الوراثة المتعددة للإنسان موجودة فى الشكل ٦-٣.

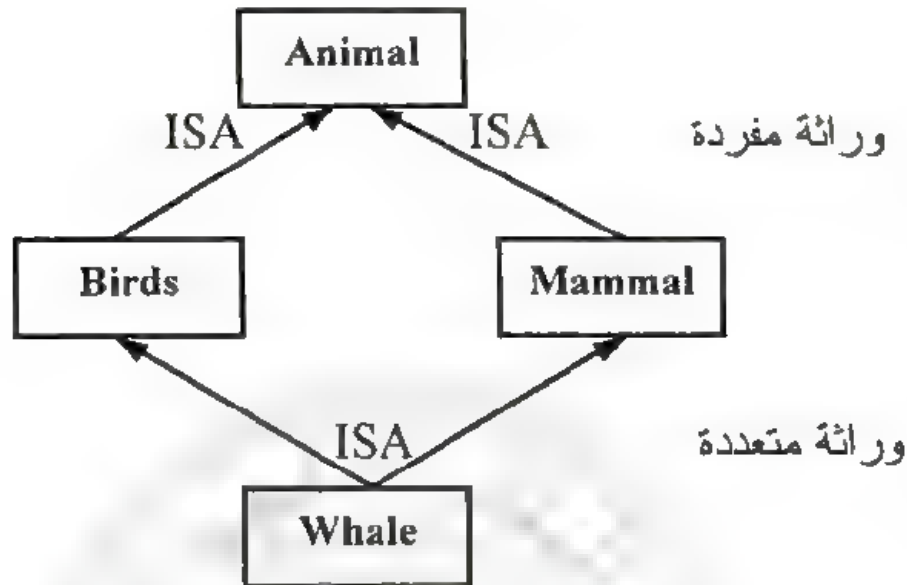


شكل ٦-٣ : شجرة وراثة متعددة للإنسان.

كما نرى فى الشكل ٦-٣، فإن كلاً من طبقة Son (ابن) و طبقة Daughter (ابنة) مشتقتان من طبقتين قاعدة هما Father (الأب) و Mother (الأم).

مثال ٣ : مطلوب بناء برنامج يعالج الوراثة المفردة و المتعددة فى جزء من المملكة الحيوانية. حيث نجد فى الشكل ٧-٣، أن الحيوان (Animal) هو الطبقة القاعدة الأولى. نشق من طبقة الحيوان طبقتين هما طبقة Birds (الطيور) و طبقة Mammal (الثدييات) كوراثة مفردة. بعد ذلك نشق من الطبقتين Birds و Mammal طبقة Whale (الحوت) التى تجمع بين صفات هاتين الطبقتين، حيث

يبيض الحوت مثل الطيور و يقوم بإرضاع صغاره بعد خروجهم من البيض مثل الثدييات. شجرة الوراثة الخاصة بهذه الطبقات نراها فى الشكل ٣-٧.



شكل ٣-٧ : شجرة وراثه من المملكة الحيوانية.

الحل :

علينا أن نعرف الطبقات الأربعة Animal و Birds و Mammal و Whale. مع الربط بين كل من Birds و Mammal كطبقات مشتقة و Animal كطبقة قاعدة. ثم نربط بين الطبقة Whale كطبقة مشتقة و الطبقتين Birds و Mammal كطبقات قاعدة.

بعد ذلك نعرف المتغيرين الخاصين aclass (فئة أو سلالة الحيوان) و legs (عدد أرجل الحيوان). كما نعرف المتغير eggs (كمية البيض فى العام) و كذلك المتغير الخاص breast (عدد ثدى الإرضاع) و نستخدم دالتين بنفس الاسم Breast مع اختلاف الأداء (تحميل زائد لأسماء الدوال). أخيراً عرفنا المتغير weight (وزن الحوت). (لاحظ أننا استخدمنا اسم المتغير aclass بدل كلمة class لأنها كلمة محجوزة).

أى كائن ينتمى إلى طبقة الحوت يمكن معالجة خمس متغيرات له و هى
 aclass و legs و eggs و breast و weight. ذلك لأن طبقة Whale ترث
 جميع خصائص طبقة Animal من خلال وراثة مفردة و طبقة Birds و طبقة
 Mammal من خلال وراثة متعددة. البرنامج الموجود فى الشكل ٣-٨ يوضّح
 تلك الأفكار.

```
#include <iostream.h>
#include <string.h>
// طبقة الحيوان العامة
class Animal {
public :
    Animal ( ) { }
    void Setclass (char *ac){strcpy(aclass, ac);}
    char *Getclass() {return aclass;}
    void Setlegs (int L) {legs = L;}
    int Getlegs ( ) {return legs;}
private:
    char aclass [30]; // رتبة أو السلالة
    int legs; // عدد الأرجل
};
// طبقة الطيور مُشتقة من طبقة الحيوان العامة
class Birds: virtual public Animal {
public :
    Birds ( ) { }
    void Seteggs (int e){eggs = e;}
    int Geteggs ( ) {return eggs;}
private :
    int eggs; // كمية البيض فى العام
};
```

// طبقة الثدييات مُشتقة من طبقة الحيوان العامة

```
class Mammal: virtual public Animal {
public :
    Mammal ( ) { }
    void Breast (int b) {breast = b;}
    int Breast( ) {return breast;}
private :
    int breast; // عدد ثدي الإرضاع
};
```

// طبقة الحوت مُشتقة من طبقتي الطيور و الثدييات

```
class Whale: public Mammal, public Birds {
public :
    Whale ( ) { }
    void Setweight (float w) {weight = w;}
    float Getweight ( ) {return weight;}
private :
    float weight; // وزن الحوت المسجل
};
```

// البرنامج الرئيسي

```
int main ( )
{
    Whale wh; // تعريف كائن حوت
    int L, e, b;
    float w;
    char c[30];

    // إدخال بيانات كائن حوت
    cout << "\n Class      : "; cin >> c;
    cout << "\n No of Legs : "; cin >> L;
    cout << "\n No of Eggs  : "; cin >> e;
```

```

cout << "\n No of breast : "; cin >> b;
cout << "\n Weight      : "; cin >> w;
wh.Setclass(c);
wh.Setlegs(L);
wh.Seteggs(e);
wh.Breast(b);
wh.Setweight(w);

```

طباعة بيانات كائن حوت //

```

cout << "\n Class      : " << wh.Getclass( );
cout << "\n No of Legs : " << wh.Gelegs( );
cout << "\n No of Eggs : " << wh.Geteggs( );
cout << "\n No of breast : " << wh.Breast( );
cout << "\n Weight      : " << wh.Getweight( ) << "\n";

```

```

return 0;

```

```

}

```

شكل ٣-٨ : برنامج معالجة الوراثة المفردة و المتعددة في المملكة الحيوانية.

لاحظ أننا استخدمنا الكلمة virtual عند اشتقاق الطبقة Birds من الطبقة Animal و كذلك فعلنا عند اشتقاق الطبقة Mammal من الطبقة Animal. يُطلق على ذلك الاشتقاق التخيلي. تفسير و أهمية ذلك نعرفه في الجزء التالي.

٣-٤ : الاشتقاق العام و الخاص و التخيلي

Public, Private, and Virtual Derivation

عند اشتقاق طبقة في كافة أشكال الوراثة التي درسناها في هذا الفصل - سواء كانت وراثة مفردة من طبقة واحدة أو وراثة متعددة من عدة طبقات - استخدمنا محدد التناول public أمام اسم الطبقة القاعدة. بالإمكان أيضاً أن نستخدم محدد التناول private أمام اسم الطبقة القاعدة.

استخدام public أو private يحدد طبيعة التناول الخارجى للطبقة بالكامل. فالطبقة التى يمكن تناولها من قبل أى دالة فى البرنامج تكون طبقة مشتقة اشتقاقاً عمومياً. أما الطبقة المحدودة التناول فهى طبقة مُشتقة اشتقاقاً خصوصياً.

■ الاشتقاق العام (public derivation)

عند اشتقاق طبقة معينة من طبقة أب باستخدام public فإن هذه الطبقة الجديدة تكون فى البداية نسخة من الطبقة الأب و لكن باسم جديد. بعد ذلك نستطيع إذا أردنا أن نضيف أعضاء تخص الطبقة الجديدة المشتقة أو نعدل من أحد أعضاء الطبقة الأب التى ورثتها بإعادة تعريفه بشكل مغاير.

■ الاشتقاق الخاص (private derivation)

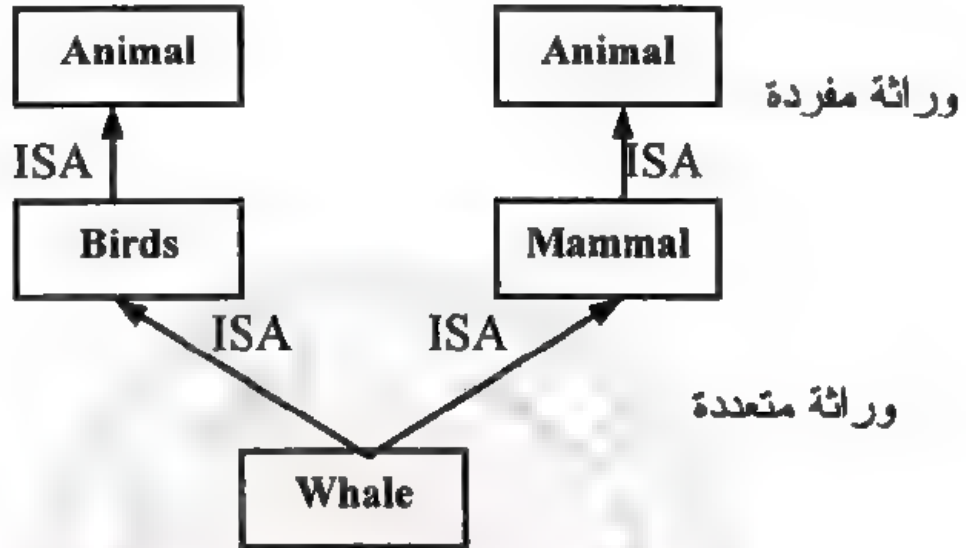
عند اشتقاق طبقة معينة من طبقة أب باستخدام private فإن الطبقة الجديدة المشتقة ترث أعضاء الطبقة الأب. لكنها ترث أعضاء الطبقة الأب من النوع العام public أو المحمى protected و يصبحوا من النوع خاص private بالنسبة للطبقة المشتقة.

■ الاشتقاق التخيلى (virtual derivation)

استخدمنا الكلمة virtual مع الاشتقاق العام public عند اشتقاق كلاً من الطبقَتَين Birds و Mammal من الطبقة Animal. هذا الاشتقاق يُطلق عليه الاشتقاق العام التخيلى (virtual public Derivation). هذا النوع من الاشتقاق نستخدمه مع الوراثة المتعددة.

لاحظ أنه حسب تعريف شجرة الوراثة فى البرنامج الموجود فى الشكل ٣-٨ يمكن أن يُفسَّر طبقاً لشجرة الوراثة الموجودة فى الشكل ٣-٩ و ليست الموجود فى الشكل ٣-٧. لكن باستخدام الكلمة virtual تم توجيه

المترجم إلى الشكل ٣-٧ و أن الطبقة Animal واحدة فقط. ذلك يمنع اللبس و الغموض داخل البرنامج عند اشتقاق كائن من الطبقة Animal، حيث يتم اشتقاق كائن واحد بدل اثنين.



شكل ٣-٩ : شجرة وراثه من المملكة الحيوانية بدون الاشتقاق التخلي.

Protected Variables

٣-٥ المتغيرات المحمية

كما نعلم أن عند تعريف متغيرات أعضاء في طبقة معينة من النوع private، لا يمكن تناول تلك المتغيرات إلا من خلال دوال معالجة تلك المتغيرات الأعضاء في نفس الطبقة. و عندما نشق طبقة أخرى من تلك الطبقة فإنها لا تستطيع تناول تلك المتغيرات الخاصة.

أما المتغيرات المحمية (protected) فهي مثل المتغيرات الخاصة (private) لا يمكن التعامل معها إلا من خلال دوال معالجتها الأعضاء في نفس الطبقة. لكن عند اشتقاق طبقة أخرى من تلك الطبقة فإن الدوال الأعضاء في الطبقة الجديدة المشتقة يمكنها تناول المتغيرات المحمية في الطبقة القاعدة.

لذلك فإن المتغيرات المحمية تعتبر في منتصف المسافة بين المتغيرات العامة و الخاصة. حيث نستطيع تناول و معالجة البيانات العامة من أى مكان. بينما المتغيرات الخاصة من خلال دوال نفس الطبقة. نستخدم المتغيرات المحمية مع الوراثة و اشتقاق طبقات من طبقة قاعدة.

الطبقة المشتقة ترث جميع المتغيرات العامة و المحمية من الطبقة الأب أو القاعدة. أى أن المتغيرات العامة الموروثة كأنها أعضاء عامة فى الطبقة المشتقة بينما المتغيرات المحمية الموروثة كأنها أعضاء خاصة فى الطبقة المشتقة.

فلو فرضنا أن لدينا طبقة ABC تضم ثلاث متغيرات الأول A من النوع public و الثانى B من النوع protected و الثالث C من النوع private. فإننا نتعامل مع المتغير A مباشرة و المتغير B و C من خلال دوال معالجتهم. و عندما نشق طبقة فرعية XYZ من الطبقة ABC، فإن الطبقة XYZ تستطيع تناول المتغير A و B مباشرة بينما لا تستطيع تناول المتغير C أبداً.

مثال ٤ : مطلوب كتابة برنامج يعرف طبقة قاعدة Disk (قرص تخزين) تضم ثلاث متغيرات أحدهما Size من النوع public و الثانى Tracks (عدد مسارات التخزين) من النوع protected و الثالث speed (سرعة التناول) من النوع الخاص. يتفرع من الطبقة Disk طبقة أخرى هى CD (قرص الليزر المدمج) تضم متغير Type (نوع القرص المدمج) من النوع public. بين كيفية معالجة بيانات كائنات الطبقتين باستخدام المؤشرات و بدونها.

الحل :

```

#include <iostream.h>
#include <string.h>
// طبقة القرص
class Disk {
public :
    Disk( ) { }
    Disk (int si, int tr, int sp)
        { Size = si; Tracks = tr; Speed =sp; }
    void SetTracks(int tr) { Tracks = t; } // إدخال المسارات
    int GetTracks ( ) {return Tracks; } // استرجاع المسارات
    void SetSpeed(int si) {Speed = s; } // إدخال السرعة
    int GetSpeed( ) {return Speed; } // استرجاع السرعة
    int Size; // الحجم
protected :
    int Tracks; // المسارات
private:
    int Speed; // السرعة
};
// طبقة القرص المدمج المشتقة
class CD: public Disk {
public:
    CD ( ) { }
    CD (int si, int tr, int sp ,char *ty);
    /* حساب كثافة التخزين باستخدام الحجم (عام) و المسارات (محمي) من
        */
        /* الطبقة القاعدة القرص
    double Density ( ) {return double(Size/Tracks);}
    char type [10];
};

```

دالة بناء القرص المدمج تستدعي دالة بناء القرص و تمرر لها المتغيرات الثلاثة /*

*/ الأوائل و تعالج المتغير الرابع فقط

```
CD::CD (int si, int tr, int sp ,char *ty):Disk(si, tr,sp)
{
    strcpy (type , ty);
}
```

//دالة البرنامج الرئيسية

```
int main ( )
```

```
{
    Disk d1, d2 (1400, 100 , 300), *dptr;
    CD cd1, cd2 (64000, 400, 1000, "RW" ), *cdptr;
    // معالجة الكائنات بدون استخدام المؤشرات
    // طباعة بيانات كائن قرص
    cout << "\n Size : " << d2.Size << " KB";
    cout << "\n Tracks : " << d2.GetTracks( );
    cout << "\n Speed : " << d2.GetSpeed( );
    // طباعة بيانات كائن قرص مدمج
    cout << "\n Size : " << cd2.Size << " KB";
    cout << "\n Tracks : " << cd2.GetTracks( );
    cout << "\n Speed : " << cd2.GetSpeed( );
    cout << "\n Density : " << cd2.Density( );
    // معالجة الكائنات باستخدام المؤشرات
    dptr = &d2;
    cdptr = &cd2;
    // طباعة بيانات كائن قرص
    cout << "\n Size : " << dptr->Size << " KB";
    cout << "\n Tracks : " << dptr->GetTracks( );
    cout << "\n Speed : " << dptr->GetSpeed( );
    // طباعة بيانات كائن قرص مدمج
    cout << "\n Size : " << cdptr->Size << " KB";
    cout << "\n Tracks : " << cdptr->GetTracks( );
```

```

cout << "\n Speed : " << cdptr->GetSpeed( );
cout << "\n Density : " << cdptr->Density( ) << "\n\n";
return 0;
}

```

شكل ٣-١٠ : معالجة المتغيرات الموروثة العامة و الخاصة و المحمية.

٣-٦ الدوال التخيلية و تعدد الأشكال

Virtual Functions and Polymorphism

عندما نكتب دالة عضو في طبقة قاعدة و نعرف مقدماً أن طبقة ما سوف نشقها من الطبقة القاعدة و تحتوى على دالة بنفس الاسم و نفس المتغيرات المرسله و المتغير الذى ترجع به الدالة، لابد أن تكون الدالة العضو في الطبقة القاعدة من النوع virtual بالشكل :

```
virtual void DisplayData ( ){cout << "\n Data";}
```

عندما لا نضع الكلمة virtual قبل اسم دالة طبقة القاعدة، لا نستطيع تنفيذ تلك الدالة المعروفة للطبقة المشتقة باستدعائها من خلال مؤشر إلى الطبقة الأب لأن المترجم سوف ينفذ النسخة الخاصة بالطبقة الأب. عند وضع دالة تخيلية في برنامج ما، فإن المترجم يُخزّن الدالة في موقع خاص بالذاكرة. ثم يتناول الدالة من خلال جدول مؤشرات.

تعدد الأشكال (polymorphism)

الدوال التخيلية هي مفتاح تعدد الأشكال. فعندما نمدعى دالة عضو تخيلية من خلال مؤشر إلى طبقة قاعدة، فإن المترجم ينفذ النسخة المعروفة في الطبقة المشتقة بدلاً من استدعاء النسخة المعروفة للطبقة القاعدة. هذا معاكس تماماً لما يحدث عند عدم استخدام الكلمة virtual.

عند الإعلان عن دالة تخيلية في طبقة قاعدة، فإن تعريف الدالة الموجود في الإعلان عن الطبقة القاعدة لا يُستخدم عادة. لكننا نستطيع إجبار المترجم على استخدامه باستخدام المعامل ":" كما نرى مع الدالة () compute في مثال ٥ الموجود في الشكل ٣-١٠. فعند تعريف دالة تخيلية يتوقع المترجم أنه سيعاد تعريفها في الطبقة المشتقة.

عند كتابة دالة تخيلية، لا يجبرنا المترجم على إعادة تعريف نفس الدالة في الطبقة المشتقة و لا يتسبب ذلك في أى مشكلة. مع أن الدوال التخيلية تُعد ميزة قوية في لغة ++C، إلا أننا لا نستطيع تصميم جميع الدوال من النوع التخيلي. يرجع هذا لأن استخدام الدوال التخيلية يسبب عبئاً على البرنامج لأن استخدام تلك الدوال يتم بطريقة غير مباشرة مما يؤدي إلى بطء التنفيذ نوعاً ما.

٣-٧ تطبيق عام محلول

مثال ٥ : تأمل شجرة الوراثة لطبقات الموظفين و أجورهم الموجودة في الشكل ٣-١١. حيث تضم شجرة الوراثة أربع طبقات : الطبقة الجذر الأولى emp (طبقة الموظف العامة) التى تضم البيانات الأساسية للموظف مثل no (رقم الموظف) و name (اسم الموظف) و هما من النوع private.

بعد ذلك نشق طبقة wageemp (الموظف الأجير) من طبقة emp اشتقاقاً عاماً. تحتوى طبقة wageemp على المتغيرين الخاصين hours (عدد ساعات العمل) و wage (أجر الساعة). هدف هذه الطبقة هو طباعة بيانات أى موظف أجير مع حساب و طباعة راتبه بدالة () float compute :-

الراتب الشهري للموظف الأجير = عدد ساعات العمل × أجر الساعة

$$\text{Salary} = \text{hours} * \text{wage}$$

بنفس الكيفية نشق طبقة manger (المدير) من طبقة emp اشتقاقاً عاماً. تحتوى طبقة manager على المتغير الخاص salary (الراتب الشهري). هدف هذه الطبقة هو طباعة بيانات و راتب أى مدير بدالة () float compute .

ثم نشق طبقة salesperson (مندوب المبيعات) من طبقة wageemp اشتقاقاً عاماً. تحتوى طبقة salesperson على المتغيرين الخاصين salesmade (حجم البيع) و commission (عمولة البيع). هدف هذه الطبقة هو طباعة بيانات أى مندوب مبيعات مع حساب و طباعة راتبه بدالة () float compute كما يلي :

الراتب الشهري لمندوب المبيعات = عدد ساعات العمل × أجر الساعة +
حجم البيع × العمولة.

$$\text{Salary} = \text{hours} * \text{wage} + \text{salesmade} * \text{commission}$$



شكل ٣-١١ : شجرة طبقات الموظفين و أجورهم.

الحل :

البرنامج الموجود فى الشكل ٣-١٢ يضم بين صفتيه تعاريف الطبقات و اشتقاقها و معالجة بياناتها و اشتقاق كائنات و مصفوفة كائنات بطريقة النقطة و اشتقاق كائنات بطريقة المصفوفات من جميع الطبقات. تأمل الدالة () compute التى استخدمناها فى جميع الطبقات لحساب راتب أى كائن موظف ينتمى إلى أى طبقة من طبقات شجرة الموظفين.

طبيعى أن الدالة () compute التى تحسب راتب كائن من طبقة موظف أجير ترثها كائنات طبقة مندوب المبيعات، لكننا عرفنا دالة تحمل نفس الاسم و المتغيرات المرسله و العائده مع اختلاف الأداء فى الطبقة salesperson و طبقة المدير لذلك استخدمنا الكلمة virtual قبل اسم الدالة.

```
#include<iostream.h>
```

```
#include<string.h>
```

```
// طبقة عامة للموظف تضم البيانات الأساسية : الرقم و الاسم
```

```
class emp{
```

```
public:
```

```
    emp( ) { /*cout<<"\nEmpty Employee constructor\n";*/ }
```

```
    emp(int a, char *b);
```

```
    void setno(int a) {no=a;} // إدخال رقم الموظف
```

```
    int getno( ) {return no;} // استرجاع رقم الموظف
```

```
    void setname (char *b) // إدخال اسم الموظف
```

```
        {strcpy(name, b);}
```

```
    char *getname( ) {return name;} // استرجاع اسم الموظف
```

```
private:
```

```
    int no; // رقم الموظف
```

```
    char name[40]; // اسم الموظف
```

```
};
```

```
// دالة بناء طبقة الموظف تتبنى كيان موظف و تدخل الرقم و الاسم
```

```
emp::emp(int a, char *b)
```

```
{
    cout <<"Employee constructor";
    setno(a);
    setname(b);
}
```

طبقة الموظف الأجير تضم متغيرين هما عدد الساعات و أجر الساعة إضافة إلى المتغيرين الموروثين من أعضاء طبقة الموظف العامة و هما الرقم و الاسم //

```
class wageemp: public emp {
public:
    wageemp( ) { /*cout<<"\nEmpty wage";*/ }
    wageemp(int a, char *b, int h, float w);
    void sethours(int h) { hours=h; } // إدخال عدد ساعات العمل
    int gethours( ) { return hours; } // استرجاع عدد ساعات العمل
    void setwage(float w) { wage=w; } // إدخال أجر الساعة
    float getwage( ) { return wage; } // استرجاع أجر الساعة

    // دالة حساب الراتب
    virtual float compute( ) { return wage*hours; }

private:
    int hours; // عدد ساعات العمل
    float wage; // أجر الساعة
};
```

دالة بناء طبقة الموظف الأجير التي تستقبل قيم ٤ متغيرات، تمرر الرقم و الاسم /*
 * لدالة بناء طبقة الموظف العامة و تعالج عدد الساعات و أجر الساعة

```
wageemp::wageemp(int a, char *b, int h, float w):emp(a, b)
{
    sethours(h);
    setwage(w);
};
```

طبقة مندوب المبيعات التي تضم متغيرين جديدين هم حجم البيع و العمولة /*
 إضافة إلى المتغيرين عدد الساعات و أجر الساعة من طبقة الموظف الأجير
 * والمتغيرين رقم الموظف و اسم الموظف من طبقة الموظف العامة

```
class salesperson: public wageemp {
public:
```

```

salesperson( )
    { /*cout<<"\nsalesperson empty constructor";*/}
salesperson(int a, char *b, int h, float w, int s, float c);
void setsalesmade(int s){salesmade=s;} //إدخال حجم البيع
int getsalesmade ( ) {return salesmade;} //استرجاع حجم البيع
//إدخال العمولة
void setcommission (float c){commission=c;}
//استرجاع العمولة
float getcommission( ){return commission;}
float compute( ) //دالة حساب الراتب
    {return wageemp::compute( )+salesmade*commission;}
private:
    int salesmade; //حجم البيع
    float commission; //العمولة
};
//دالة بناء طبقة مندوب المبيعات التى تستقبل قيم ٦ متغيرات، تمرر ٤ منها إلى دالة *
//بناء طبقة الموظف الأجير ( رقم الموظف و اسم الموظف و عدد الساعات و
// *أجر الساعة) و تعالج هى متغيرين من طبقتهما و هما حجم البيع و العمولة
salesperson::salesperson(int a,char *b,int h,float w,int s,float c):
    wageemp(a, b, h, w)
{
    setsalesmade(s);
    setcommission(c);
    cout <<"\nSales person full constructor";
}

//طبقة المدير التى تضم متغير واحد هو الراتب إضافة إلى *
//المتغيرين الموروثين من طبقة الموظف العامة و هما الرقم و الاسم
class manger: public emp{
public:
    manger( ){/*cout<<"\nmanger empty constructor";*/}
    manger(int a, char *b, float sal);
    void setsalary (float sal){salary=sal;} //إدخال الراتب الشهرى
    //استرجاع الراتب الشهرى
    float compute( ){return salary;}

```


private:

float salary;// الراتب الشهري

};

دالة بناء كائن من طبقة المدير تستقبل ٣ متغيرات، تمرر متغيرين هما الرقم و
 /* الاسم لطبقة الموظف العامة و تعالج متغير طبقتها الراتب الشهري

manger::manger(int a, char *b, float sal):emp(a, b)

{

setsalary(sal);

cout <<"\nManger full constructor";

}

int main()

{

// معالجة بيانات كائنات طبقة الموظف العامة

emp e1(425,"Sayf"), e2, e5[20], *eptr;

eptr = &e2;

// طباعة بيانات كائن موظف

cout <<"\n Employee Number : " <<e1.getno();

cout <<"\n Employee Name : " <<e1.getname();

int y,i;

char z[10];

cout <<"\n Enter No, Name : \n";

cin >>y >>z;

// معالجة بيانات الكائن باستخدام المؤشرات

eptr->setno(y);

eptr->setname(z);

cout <<"\n Employee Number : " <<eptr->getno();

cout <<"\n Employee Name : " <<eptr->getname();

// معالجة بيانات مصفوفة كائنات من طبقة الموظف العامة

int n;

cout <<"\nEnter No of Employee : "; cin >> n;

إدخال بيانات مصفوفة كائنات طبقة الموظف العامة//

```
for(i=0; i<n; i++)
```

```
{
```

```
    cout <<"\n Enter No, Name : \n";
```

```
    cin >>y >>z;
```

```
    e5[i].setno(y);
```

```
    e5[i].setname(z);
```

```
};
```

إدخال بيانات مصفوفة كائنات طبقة الموظف العامة//

```
for(i=0; i<n; i++)
```

```
{
```

```
    cout <<"\n Employee Number : " <<e5[i].getno( );
```

```
    cout <<"\n Employee Name : " <<e5[i].getname( );
```

```
    cout<<"\n";
```

```
}
```

معالجة بيانات كائنات طبقة الموظف الأجير//

```
wageemp w1, w2(427,"Alla",50,130.5),w5[20], *wptr;
```

```
wptr = &w1;
```

طباعة بيانات كائن من طبقة الموظف الأجير//

```
cout <<"\nEmployee Number : " <<w2.getno( );
```

```
cout <<"\nEmployee Name: " <<w2.getname( );
```

```
cout<<"\nSalary : " <<w2.compute( );
```

```
int h, a;
```

```
float w;
```

```
char k[20];
```

```
cout <<"\n Enter No, Name , hours, wage : \n";
```

```
cin >>a >>k >>h >>w;
```

معالجة بيانات كائن من طبقة الموظف الأجير باستخدام المؤشرات//

إدخال بيانات كائن من طبقة الموظف الأجير//

```
wptr->setno(a);
```

```
wptr->setname(k);
```

```
wptr->sethours(h);
```

```
wptr->setwage(w);
```

طباعة بيانات كائن من طبقة الموظف الأجير//

```
cout<<"\nEmployee Number : " <<wptr->getno( );
```

```

cout << "\nEmployee Name: " << wptr->getname( );
cout << "\nSalary: " << wptr->compute( );
// معالجة بيانات مصفوفة كائنات من طبقة الموظف الأجير
// إدخال بيانات مصفوفة كائنات من طبقة الموظف الأجير
cout << "\n Enter Number of Wage Employee : "; cin >> n;
for(i=0; i<n; i++)
{
    cout << "\n Enter No, Name , hours, wage : \n";
    cin >> a >> k >> h >> w;
    w5[i].setno(a);
    w5[i].setname(k);
    w5[i].sethours(h);
    w5[i].setwage(w);
};

// طباعة بيانات مصفوفة كائنات من طبقة الموظف الأجير
for(i=0; i<n; i++)
{
    cout << "\nEmployee Number : " << w5[i].getno( );
    cout << "\nEmployee Name: " << w5[i].getname( );
    cout << "\nSalary: " << w5[i].compute( );
};

// معالجة بيانات مندوب المبيعات
cin >> n;
int x1, x2, x3;
float y1, y2;
char q[50];
salesperson s2(427, "Doaa", 20, 130.5, 20, 1), s1, s3[20], *sptr;
sptr = &s1;
// طباعة بيانات كائن من طبقة مندوب المبيعات
cout << "\nEmployee Number : " << s2.getno( );
cout << "\nEmployee Name : " << s2.getname( );
cout << "\nSalary: " << s2.compute( ) << "\n";

```

```

// معالجة بيانات كائن من طبقة مندوب المبيعات باستخدام المؤشرات
cout<<"\nNo,Name,hours,wage,salesmade,commission:\n";
cin >>x1 >>q >>x2 >>y1 >>x3 >>y2;
// إدخال بيانات كائن
sptr->setno(x1);
sptr->setname(q);
sptr->setwage(y1);
sptr->sethours(x2);
sptr->setsalesmade(x3);
sptr->setcommission(y2);
// طباعة بيانات كائن
cout<<"\nEmployee Number : "<<sptr->getno( );
cout <<"\nEmployee Name : "<<sptr->getname( );
cout <<"\nSalary: " <<sptr->compute( )<<"\n";

// معالجة بيانات مصفوفة كائنات من طبقة مندوب المبيعات
// إدخال بيانات مصفوفة كائنات من طبقة مندوب المبيعات
for(i=0;i<2;i++)
{
cout<<"\nNo,Name,hours,wage,salesmade,commission:\n";
cin >>x1 >>q >>y1 >>x2 >>x3 >>y2;
s3[i].setno(x1);
s3[i].setname(q);
s3[i].sethours(x2);
s3[i].setwage(y1);
s3[i].setsalesmade(x3);
s3[i].setcommission(y2);
};
// طباعة بيانات مصفوفة كائنات من طبقة مندوب المبيعات
for(i=0;i<1;i++)
{
cout << "\nEmployee Number : " <<s3[i].getno( );
cout << "\nEmployee Name : " <<s3[i].getname( );
cout <<"\nEmployee Salary : " <<s3[i].compute( );
cout <<"\n\n";
}

```

```
};
```

```
// معالجة بيانات كائنات طبقة المدير
```

```
cin>>n;
```

```
int x11;
```

```
float y11;
```

```
char q1[50];
```

```
manger m1, m2(429,"Ahmad",12500), m3 [20], *mptr;
```

```
mptr = &m1;
```

```
// طباعة بيانات كائن من طبقة المدير
```

```
cout<<"\nManager Number : " <<m2.getno( );
```

```
cout<<"\nManager Name: " <<m2.getname( );
```

```
cout<<"\nManager Salary : " <<m2.compute( );
```

```
// معالجة بيانات كائن من طبقة المدير باستخدام المؤشرات
```

```
cout<<"\n Enter No, Name, Salary :\n";
```

```
cin >>x11 >>q1 >>y11;
```

```
// إدخال بيانات كائن مدير
```

```
mptr->setno(x11);
```

```
mptr->setname(q1);
```

```
mptr->setsalary(y11);
```

```
// طباعة بيانات كائن مدير
```

```
cout <<"\nManager Number : " <<mptr->getno( );
```

```
cout <<"\nManager Name : " <<mptr->getname( );
```

```
cout <<"\nManager Salary : " <<mptr->compute( );
```

```
cout <<"\n\n";
```

```
// معالجة بيانات مصفوفة كائنات طبقة المدير
```

```
// إدخال بيانات مصفوفة كائنات طبقة المدير
```

```
for(i=0;i<2;i++)
```

```
{
```

```
    cout<<"\n Enter No, Name, Salary :\n";
```

```
    cin >>x11 >>q1 >>y11;
```

```
    m3[i].setno(x11);
```

```
    m3[i].setname(q1);
```



```

        m3[i].setsalary(y11);
    };
    // طباعة بيانات مصفوفة كائنات طبقة المدير
    for(i=0;i<2;i++)
    {
        cout<<"\nManager Number : " <<m3[i].getno( );
        cout<<"\nManager Name: " <<m3[i].getname( );
        cout<<"\nManager Salary : " <<m3[i].compute( );
    };
    return 0;
}

```

شكل ٣-١٢ : برنامج معالجة كائنات و مصفوفات الكائنات لطبقات شجرة الموظفين باستخدام الدوال التخيلية و المؤشرات و بدون المؤشرات.

٣-٨ أسئلة

١. أكتب برنامجاً يعرف class للكتاب BOOK له متغيرين خاصيين رقم الكتاب Bno و دار النشر Pub. يتفرع منه class لكتاب الحاسب Computer له متغيرين من النوع Private هما اسم الكتاب Bname و اسم المؤلف Author. ثم يقوم البرنامج بتعريف Object من كل نوع باستخدام مؤشر مع ملاً كلاً منهم بالبيانات.

٢. أكتب برنامجاً يعرف class لشخص به متغير واحد خاص هو اسم الشخص و يتفرع منه class للمواطن يحتوى على رقم بطاقة الأحوال. على أن يقوم البرنامج بإنشاء مصفوفة للمواطن و ملاًها بالبيانات ثم طباعتها.

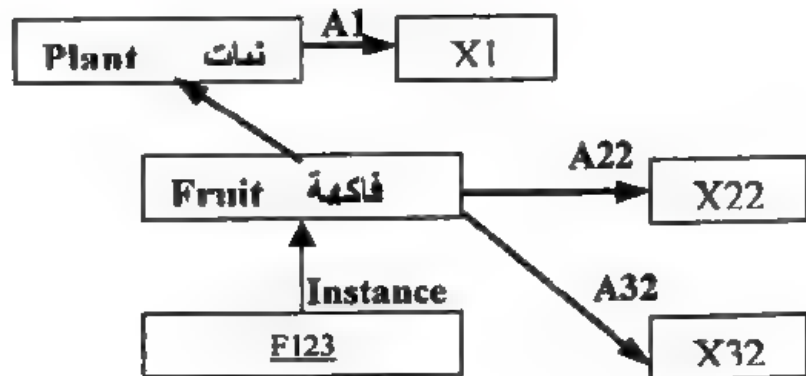
٣. أكتب برنامجاً يعرف class لشخص به متغير واحد خاص هو اسم الشخص و يتفرع منه class للمقيم يحتوى على رقم الإقامة. على أن يقوم البرنامج بإنشاء مصفوفة للمقيم و ملأها بالبيانات ثم طباعتها.

٤. أكتب البرنامج الذى يعالج بيانات كائنات طلبة الجامعة حسب شجرة الوراثة الموجودة فى الشكل ٣-١. افترض متغيرات خاصة و استخدم المؤشرات.

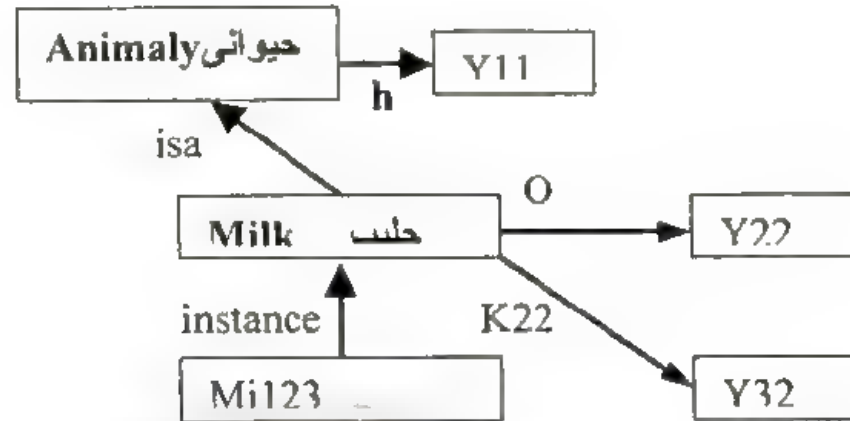
٥. أكتب برنامجاً لمعالجة بيانات كائنات اللغات الرياضية على أساس طبقة جذر للعبة ثم يتفرع منها ثلاث لغات. افترض متغيرات عامة و عالج مصفوفة كائنات.

٦. اكتب برنامجاً يعرف class لشخص به متغير واحد خاص هو اسم الشخص و يتفرع منه class للمواطن يحتوى على رقم بطاقة شخصية. و يقوم البرنامج بإنشاء مصفوفة للمواطن و ملأها بالبيانات ثم طباعتها.

٧. أكتب برنامجاً بلغة C++ لتمثيل شجرة المعرفة التالية، ثم اجعل البرنامج يدخل و يستخرج بيانات العديد من نماذج الفاكهة.



٨. أكتب برنامجاً بلغة ++C لتمثيل شجرة المعرفة التالية، ثم اجعل البرنامج يدخل و يستخرج بيانات العديد من نماذج الفاكهة.



الفصل الرابع

لغة المنطق الرياضي

Mathematical Logical Language (First Order Logic)

عَلِّمْنَا فِي الْفصل الثَّانِي أَن تَمثِيل المَعْرِفَة بِاسْتِخْدَام المنطق الرياضي أَوْ مَا يُعْرَف بِلُغَة المنطق الرياضي (Mathematical Logical Language) يَعْتَمِد عَلَى أَن المَعْرِفَة تَأْخُذ أَحَدَ شَكْلَيْنِ هُمَا : الْحَقَائِق (Facts) وَ الْقَوَاعِد (Rules). نَطْلُق أحياناً عَلَى هَذَا التَّمثِيل اسْمَ التَّمثِيل الرياضي المنطقي (Mathematical Logical Representation) أَوْ منطق المُسَنَد (Predicate Logic) أَوْ منطق الدرجة الأولى (First Order Logic).

٤-١ تمثيل الحقائق و القواعد

Representing Facts and Rules

الْحَقَائِق (Facts) هِيَ مَعْلُومَة نَصِف شَيْءً بِذَاتِهِ وَتَخْصُهُ (اسْمِيَّة أَوْ فَعْلِيَّة أَوْ شَبَه جُمْلَة). أَمَّا الْقَوَاعِد (Rules) وَ هِيَ مَعْلُومَة يُمْكِن تَعْمِيمُهَا أَوْ تَطْبِيقُهَا عَلَى مَجْمُوعَة مِنَ الْأَشْيَاءِ مِثْل (مَنْ يَذَاكِرُ يَنْجَحُ) وَ يُلْزَمُ لِتَطْبِيقِهَا أَوْ تَعْمِيمِهَا تَوْفُّرُ شَرْطٍ أَوْ مَجْمُوعَة شُرُوطٍ.

يتم تمثيل الحقائق و القواعد فى لغة المنطق الرياضى باستخدام المُسند

(predicate). و الشكل العام للمسند هو :

Predicate_Name (Argumen 1, ..., Argument_n)

XYZ(A, B) : مثال

حيث اسم المسند (Predicate Name) هو ABC و يمكن أن يكون اسماً أو فعلاً. أما العناصر بين القوسين (arguments) فهما (A و B) وهى المتغيرات أو الثوابت الذين يشملهم المسند و نستخدم المسند فى تمثيل الحقيقة بينما يمكن أن يتم تمثيل قاعدة معينة باستخدام مسندين أو أكثر. مع ملاحظة أن ما بداخل الأقواس يكون قيم ثوابت عند تمثيل الحقيقة. أما عند تمثيل القاعدة فيكون واحداً على الأقل مما بين الأقواس متغيراً حتى يمكن التعويض عنه.

٤-١-١ قواعد التحويل إلى التمثيل الرياضى المنطقى

عند تحويل جمل (صيغ) لغوية إلى تمثيل آخر (مثل التمثيل الرياضى المنطقى) يجب أن نقرر أولاً أياً منها حقائق (facts) و أياً منها قواعد (rules) قبل تمثيلها. فالتمثيل الجيد يؤدي إلى برنامج AI ليس صحيحاً فحسب وإنما سريع ودقيق. عند تحويل جمل لغوية تمثل قواعد إلى التمثيل الرياضى المنطقى نستخدم بعض الرموز المتعارف عليها و الموجودة فى جدول ٤-١.

Representing Facts

٤-١-٢ تمثيل الحقائق

فى الغالب تكون الحقيقة عبارة عن جملة اسمية (مبتدأ و خبر أى موصوف و صفة) أو جملة فعلية (فعل و فاعل و مفعول به أو حال). المهم أنها لا تحتوى على معلومات مجهولة أو غير محددة (أى متغيرات) بل تحتوى على معلومات معلومة و قيم ثوابت.

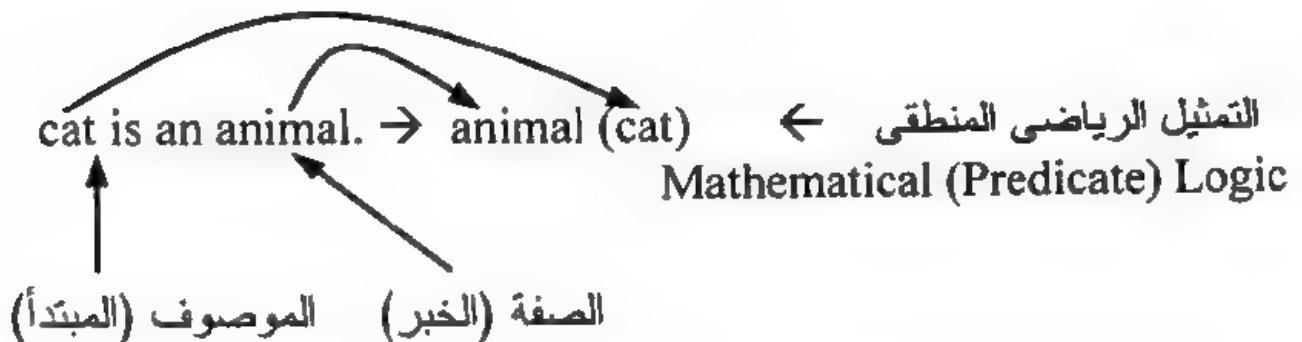
جدول ٤-١ : الرموز المُستخدَمة فى التمثيل الرياضى المنطقى

الرمز Symbol	المعنى Meaning
\forall	لكل : For All
\exists	يوجد : Exist
\neg	النفى (لا) : NOT
\wedge	و : AND
\vee	أو : OR
\rightarrow	يؤدى إلى، إذا : Then

- **الجملة الاسمية** : يتم استخدام الصفة أو الخبر اسماً للمسند و يكون الموصوف هو العنصر الموجود بداخل الأقواس، كما فى المثال الموجود فى شكل ٤-١.

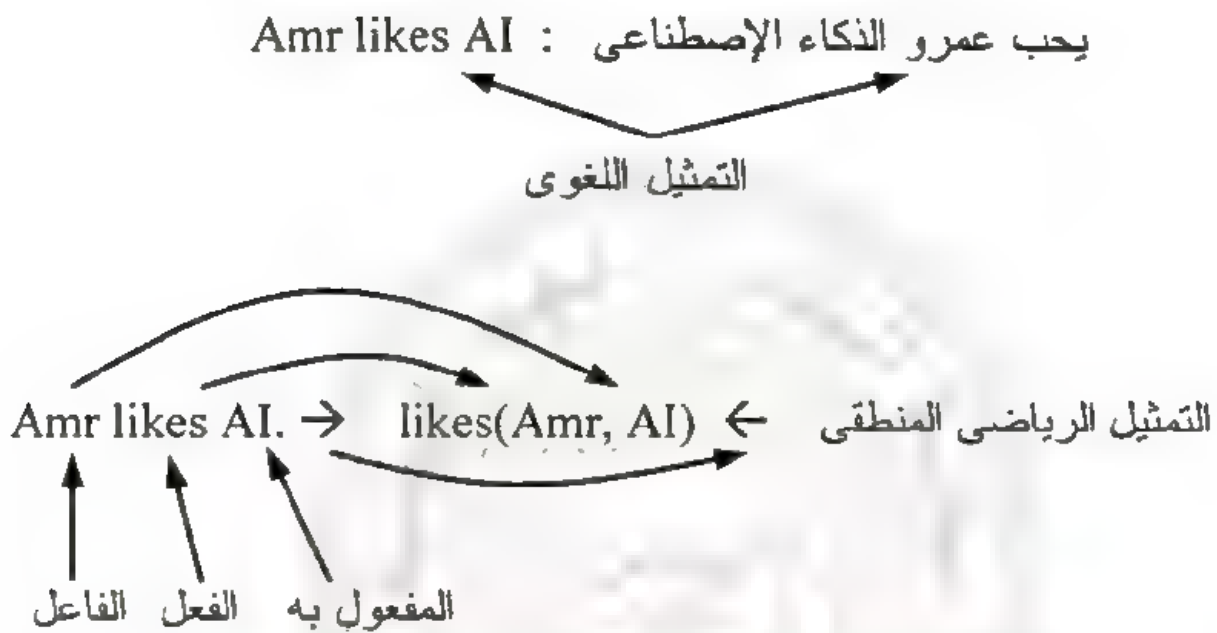
القط حيوان : القط مبدأ (موصوف) وحيوان خبر (صفة) : cat is an animal

التمثيل اللغوى



شكل ٤-١ : تمثيل حقيقة (جملة اسمية) بالتمثيل الرياضى المنطقى.

- **الجملة الفعلية :** فى الغالب تتكون الجملة الفعلية من فعل و فاعل ثم مفعول به أو حال أو شبه جملة. يتم استخدام الفعل اسما للمسند و يكون الفاعل هو أول عنصر داخل الأقواس يليه فاصلة ثم العنصر الثانى و هو المفعول به أو الحال أو شبه الجملة، كما فى المثال الموجود فى شكل ٤-٢



شكل ٤-٢: تمثيل حقيقة (جملة فعلية) بالتمثيل الرياضى المنطقى.

يُحب عمرو الذكاء الاصطناعى : يحب عبارة عن فعل و يكون اسم المسند، و عمرو عبارة عن فاعل و الذكاء الاصطناعى مفعول به و هما العنصران الموجودان بين قوسى المُسند (arguments).

- **أمثلة على تمثيل الحقائق :** شكل ٤-٣ يعرض بعض الجمل فى شكلها اللغوى الطبيعى و هى عبارة عن حقائق و تضم جمل اسمية و جمل فعلية. بينما يعرض شكل ٤-٤ التمثيل الرياضى المنطقى للحقائق اللغوية الموجودة فى شكل ٤-٣.

- 1- Azzah is a Moslem.
- 2- Sayf speaks Arabic.
- 3- Artificial intelligence is a computer course.
- 4- Mango is a food.

شكل ٣-٤ : أمثلة لجمل لغوية تمثل حقائق.

- 1- moslem(Azzah).
- 2- speaks(Sayf,Arabic).
- 3- computer_course(Artificial_intelligence).
- 4- food(Mango).

شكل ٤-٤ : التمثيل الرياضى المنطقى للحقائق

الموجودة فى شكل ٣-٤.

Representing Rules

٣-١-٤ تمثيل القواعد

فى الغالب يمكن وضع تمثيل أى قاعدة فى صورة جملة IF ... THEN الشرطية لتأخذ الشكل العام التالى :

IF (اسم الشرط) THEN (جواب الشرط)

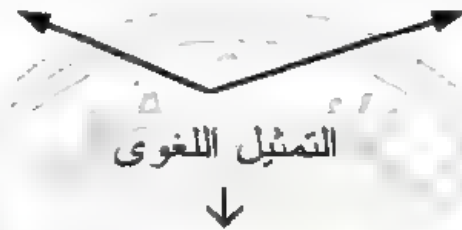
من الممكن أن يكون اسم الشرط و/أو جوابه عبارة عن جزء واحد أو أكثر. و كل جزء من هذه الأجزاء عبارة عن حقيقة ويتم تمثيلها أيضاً باستخدام مسند واحد.

مع مراعاة أنه فى الغالب تحتوى هذه الحقيقة و المسند الناتج عنها على متغير مجهول أو أكثر بالإضافة إلى وجود الثوابت. أيضاً ربما يكون أى جزء

عبارة عن جملة اسمية أو فعلية نطبق بحققها الأسلوب الذى اتبعناه مع كل منهما فيما سبق.

شكل ٤-٥ يعرض مثلاً على كيفية تحويل القاعدة إلى التمثيل الرياضى المنطقى. يعرض شكل ٤-٦ مجموعة من الجمل اللغوية التى تحتوى على قواعد يمكن تمثيلها تمثيلاً رياضياً منطقياً كما فى الشكل ٤-٧.

جميع المصريين يتكلمون العربية : All egyptians speak Arabic



أى مصرى يتكلم العربية : IF (X is egyptian) Then (X speak Arabic)



جملة اسمية

جملة فعلية

X موصوف بصفة مصرى

يتكلم فعل و X فاعل و عربى مفعول به



egyptian (X)

then

speaks(X, Arabic)



$\forall x: \text{egyptian}(X) \rightarrow \text{speaks}(X, \text{Arabic})$



التمثيل الرياضى المنطقى

شكل ٤-٥: تمثيل قاعدة بالتمثيل الرياضى المنطقى.

- 1- Ahmad likes all computer courses.
- 2- All Arab speak Arabic.
- 3- Alla eats anything Doaa eats.
- 4- All Computer students study programming languages.

شكل ٤-٦ : أمثلة لجمل لغوية تمثل قواعد.

- 1- $\forall x: \text{computer_course}(x) \rightarrow \text{likes}(\text{Ahmad}, x).$
- 2- $\forall x: \text{Arab}(x) \rightarrow \text{speaks}(x, \text{Arabic}).$
- 3- $\forall x: \text{eats}(\text{Doaa}, x) \rightarrow \text{Alla eats anything Doaa eats}.$
- 4- $\forall x: \forall y: \text{computer_student}(x) \wedge \text{programming_language}(y) \rightarrow \text{study}(x, y).$

شكل ٤-٧ : التمثيل الرياضى المنطقى للقواعد

الموجودة فى شكل ٤-٦.

٤-٢ الاستنتاج باستخدام المنطق الرياضى

Deduction using Mathematical Logic

نستطيع استخدام الجمل المُمثلة بالمنطق الرياضى فى الحصول على معرفة جديدة عن طريق الاستنتاج و استخلاص معلومات قديمة من المعلومات الموجودة حالياً. نستفيد من المعلومات الجديدة فى حل المشكلات.

٤-٢-١ خطوات التمثيل و الاستنتاج

Steps of Representing and Deduction

للاستدلال و الوصول إلى حلول نمر خلال خطوات عملية التمثيل والاستنتاج التالية:

- لدينا مجموعة من الجمل أو الصيغ اللغوية.
- يتم تمثيل هذه الجمل باستخدام لغة المنطق الرياضى.
- يمكن استنتاج حقائق جديدة باستخدام طريقة الاستنتاج الرياضى (deductive mechanism of logic).
- نبدأ من الشيء المطلوب الوصول إلى حل له و نفرض حلاً معيناً (الهدف) و نضعه فى نفس التمثيل الرياضى المنطقى.
- باستخدام طريقة التسلسل العكسى (backward chaining) نحاول إثبات صحة الفرض أو عدمه بالطريقة التالية (عمل برنامج AI):

١. نبحث فى التمثيل الرياضى للجمل اللغوية (قاعدة المعرفة المعلومة لدينا) عن حقيقة تؤكد صحة الفرض من عدمه. فإذا وجدنا ذلك نكون قد وصلنا إلى الحل.

٢. فى حالة عدم وجود الحقيقة المناسبة نبحث عن حقيقة أو قاعدة طرفها الأيمن هو نفس الفرض المطلوب إثبات صحتها (بالطبع مع إجراء بعض التعويضات).

٣. فإذا لم نجد نغير الفرض إلى الإثبات (الإجابة بنعم)، و نعيد الخطوتين ١ و ٢.

٤. عند وجود الطرف الأيمن المناسب، يكون علينا الآن إثبات صحة المُسندات التى يتكون منها الطرف الأيسر و نكرر الخطوة ب حتى نقوم بإثبات جميع المسندات و لا يتبقى أى مُسند بدون إثبات. فى هذه الحالة يكون الفرض صحيح و إلا فإن الفرض يكون خطأ.

٥. أحياناً يصل البرنامج فى النهاية إلى مُسند لا يستطيع تحديد صحته من عدمها. فى هذه الحالة يقوم البرنامج بعرض المُسند و يسأل المستخدم عن علاقته بأى من جمل التمثيل الرياضى المنطقى. و فى

حالة إضافة المستخدم معلومة يتعلمها برنامج AI و يُضيفها إلى قاعدة المعرفة (التمثيل الرياضى).

٤-٢-٢ مثال ١

لدينا مجموعة الجمل اللغوية التالية الموجودة فى شكل ٤-٨.

6. Database is a computer course.
7. Editing reports is a language course.
8. Ahmad likes difficult courses and easy courses.
9. Artificial Intelligence is a computer course.
10. Sayf study any difficult course.
11. Computer courses are not easy course.
12. Language courses are difficult.
13. Ahmad a computer student.
14. Sayf is a computer student.
15. Alla studies whatever Sayf studies.

شكل ٤-٨ : جمل لغوية لوصف دراسة المقررات الدراسية.

- المطلوب تحويل هذه الجمل اللغوية إلى التمثيل الرياضى المنطقى و الإجابة على السؤال التالى :

Does Alla study editing reports?

- للإجابة على هذا السؤال نقوم بالخطوات التالية :

- نُحوّل الجمل اللغوية إلى التمثيل الرياضى المنطقى.
- نفرض أن الإجابة على السؤال هى النفى.
- نُحوّل السؤال إلى التمثيل الرياضى المنطقى.

- باستخدام التسلسل العكسى نحاول إثبات صحة أو عدم صحة الفرض.
- عند الفشل نعود و نفرض الإجابة نعم.

■ الإجابة :

- تحويل الجمل اللغوية إلى التمثيل الرياضى المنطقى كما نرى فى الشكل ٩-٤.

1. computer_course (database).
2. language_course (editing_reports).
3. $\forall x: \text{difficult_course}(X) \vee \text{easy_course}(X) \rightarrow \text{studies}(\text{Ahmad}, X)$.
4. computer_course(artificial_intelligence).
5. $\forall x: \text{difficult_course}(X) \rightarrow \text{studies}(\text{Sayf}, X)$.
6. $\forall x: \text{computer_course}(X) \rightarrow \neg \text{easy_course}(X)$.
7. $\forall x: \text{language_course}(X) \rightarrow \text{difficult_course}(X)$.
8. computer_student(Ahmad).
9. computer_student(Sayf).
10. $\forall x: \text{studies}(\text{Sayf}, X) \rightarrow \text{studies}(\text{Alla}, X)$

شكل ٩-٤ : التمثيل الرياضى المنطقى للجمل اللغوية
الموجودة فى شكل ٨-٤.

- فرض الإجابة بالنفى على السؤال كما يلى :

No, Alla Doesn't study editing report.

- تحويل الإجابة إلى التمثيل الرياضى المنطقى كما يلى :

$\neg \text{studies}(\text{Alla}, \text{editing_report})$.

- استخدام التسلسل العكسى (لإثبات صحة الفرض أو عدمه) كما فى الشكل ٤-١٠.

studies (Alla, editing_report).



لا يوجد حقيقة أو طرف أيمن لقاعدة تتوافق مع هذا المسند

شكل ٤-١٠ : فشل الحل باستخدام التسلسل العكسى مع فرض منفى.

لذلك نغير الفرض إلى الإثبات أى (studies (Alla, editing_report) و نبدأ الحل من جديد كما فى شكل ٤-١١.

studies (Alla, editing_report).

↑ من ١٠ و بالتعويض X=editing_report

studies (Sayf, editing_report).

↑ من ٥ و بالتعويض X=editing_report

difficult_course(editing_report)

↑ من ٧ و بالتعويض X=editing_report

language_course(editing_report)

↑ من ٢

nil

شكل ٤-١١ : الحل باستخدام التسلسل العكسى مع فرض مثبت.

طالما أننا وصلنا إلى nil (لاشئ) فإن الفرض يكون صحيحاً و تكون الإجابة هى نعم تدرس آلاء تحرير التقارير أى

yes, Alla studies editing reports

٤-٢-٣ مثال ٢

إذا كان لديك مجموعة الجمل اللغوية الموجودة فى شكل ٤-١٢. المطلوب تمثيلها باستخدام المنطق الرياضى و الإجابة على السؤال التالى :

Does Amr like mango?

1. Amr likes all kinds of foods
2. Apples are foods.
3. Chicken is food.
4. anything anyone eat and isn't killed by is a food.
5. Nada eats mango and is not killed by.
6. Walid eats anything Amr eats.
7. Any food should have tasty for some persons.

شكل ٤-١٢ : جمل لغوية لوصف تناول الأطعمة.

الحل :

- تحويل الجمل اللغوية إلى التمثيل الرياضى المنطقى كما نرى فى الشكل ٤-١٣.

1. $\forall x: \text{food}(X) \rightarrow \text{likes}(\text{Amr}, X).$
2. $\text{food}(\text{apple}).$
3. $\text{food}(\text{chicken}).$
4. $\forall x: \forall y: \text{eats}(X, Y) \wedge \neg \text{killed}(X) \rightarrow \text{food}(Y).$
5. $\text{eats}(\text{Nada}, \text{Mango}) \wedge \neg \text{killed}(\text{Nada}).$
6. $\forall x: \text{eats}(\text{Amr}, X) \rightarrow \text{eats}(\text{Walid}, X).$
7. $\forall x: \exists y: \text{food}(X) \wedge \text{person}(Y) \rightarrow \text{have_tasty}(Y, X).$

شكل ٤-١٣ : التمثيل الرياضى المنطقى للجمل اللغوية

الموجودة فى شكل ٤-١٢.

- فرض الإجابة بالنفى على السؤال كما يلي :

No, Amr doesn't like mango?

- تحويل الإجابة إلى التمثيل الرياضى المنطقى كما يلي :

$\neg \text{likes}(\text{Amr}, \text{Mango}).$

- استخدام التسلسل العكسى (لإثبات صحة الفرض أو عدمه). لا نجد فى الجمل الموجودة فى شكل ٤-١٣ أى حقيقة أو طرف أيمن فى قاعدة مثل الفرض. لذلك نعكس الفرض إلى الاثبات و نبدأ الحل كما فى شكل ٤-١٤.

$\text{likes}(\text{Amr}, \text{Mango}).$

↑ من ١ و بالتعويض $X=\text{Mango}$

$\text{food}(\text{mango})$

↑ من ٤ و بالتعويض $Y=\text{Mango}$

$\text{eats}(X, \text{Mango}) \wedge \neg \text{killed}(X)$

↑ من ٥ و بالتعويض $X=\text{Nada}$

nil

شكل ٤-١٤ : الحل باستخدام التسلسل العكسى مع فرض مثبت.

طالما أننا وصلنا إلى nil (لأشياء) فإن الفرض يكون صحيحاً و تكون

الإجابة هى نعم يحب عمرو المانجو أى

yes, Amr likes Mango.

٤-٢-٤ مثال ٣

أعد حل المثال السابق فى ٤-٢-٣ بعد تغيير الجملة اللغوية رقم ١٠ إلى الشكل
التالى :

10. Alla studies any course Sayf studies.

تحويل الجملة اللغوية رقم ١٠ إلى التمثيل الرياضى المنطقى يكون كما يلى :

10. $\forall x: \text{course}(X) \wedge \text{studies}(\text{Sayf}, X) \rightarrow \text{studies}(\text{Alla}, X)$

نظراً لاختلاف الجملة اللغوية رقم ١٠ فى المثال الحالى عن تلك الموجودة فى شكل ٤-٨ فقد اختلف التمثيل الرياضى لها عن ذلك الموجود فى شكل ٤-٩. أيضاً يتغير الحل باستخدام الارتداد العكسى عما هو موجود فى شكل ٤-١١ إلى الحل الموجود فى شكل ٤-١٥ .

$\text{studies}(\text{Alla}, \text{editing_report}).$

↑ من ١٠ و بالتعويض $X=\text{editing_report}$

$\text{course}(\text{editing_report}) \wedge \text{studies}(\text{Sayf}, \text{editing_report}).$

↑ من ٥ و بالتعويض $X=\text{editing_report}$

$\text{course}(\text{editing_report}) \wedge \text{difficult_course}(\text{editing_report})$

↑ من ٧ و بالتعويض $X=\text{editing_report}$

$\text{course}(\text{editing_report}) \wedge \text{language_course}(\text{editing_report})$

↑ من ٢

$\text{course}(\text{editing_report})$

شكل ٤-١٥ : الحل باستخدام التسلسل العكسى مع فرض مُثبت

مع تعديل الجملة اللغوية رقم ١٠.

كما نرى فى شكل ٤-١٥ وصل الحل إلى مُسند معين مطلوب اثباته و هو $course(editing_report)$ و لا يستطيع الشخص أو برنامج AI اثبات صحة هذا المسند من عدمه استناداً إلى التمثيل الرياضى (قاعدة المعرفة) الموجود لدينا. مع العلم أن $editing_report$ هو $course$ و نعلم من الجمل الموجود لدينا أنه $language_course$ و أن أى $language_course$ لابد أن يكون $course$ لكن برنامج AI لا يعلم ذلك. لذا يجب أن نخبر برنامج AI بتلك المعلومة ليضيفها إلى قاعدة المعرفة الموجودة لديه. المعلومة الجديدة هى :

11. Any language course is a course

و تمثيلها الرياضى المنطقى يكون :

11. $\forall x: language_course(X) \rightarrow course(X).$

بناءً على القاعدة رقم ١١ نُكمل الحل الموجود فى شكل ٤-١٥ كما هو مبين فى شكل ٤-١٦.

↑ من ٢

$course(editing_report)$

↑ من ١١ و بالتعويض $X=editing_report$

$language_course(editing_report)$

↑ من ٢

nil

شكل ٤-١٦ : الحل باستخدام التسلسل العكسى مع فرض مُثبت

مع تعديل الجملة للغوية رقم ١٠.

٤-٢-٥ تمثيل العمليات الحسابية

Computation Function and Predicate

نستطيع تمثيل العمليات الحسابية و المنطقية داخل التمثيل الرياضى المنطقى. جدول ٤-٢ يعرض بعضاً من تلك العمليات و طريقة تمثيلها.

جدول ٤-٢ : الرموز الحسابية أو المنطقية

الرمز Symbol	المعنى Meaning
gt	greater than أكبر من '>'
lt	Less than أقل من '<'
ge	greater than or equal أكبر من أو يساوى '>='
le	less than or equal أقل من أو يساوى '<='
=	equal = يساوى

فيما يلى نعرض بعض الأمثلة لاستخدام الرموز الموجودة فى جدول ٤-٢

كمسندات فى التمثيل الرياضى المنطقى :

gt(A,B) : gt(3,1) مُسند صحيح

lt(A,B) : lt(1,3) مُسند صحيح

ge(C,D) : ge(7,9) مُسند خطأ

le(X,Y) : le(9,7) مُسند خطأ

٤-٢-٦ مثال ٤

إذا كان لديك مجموعة الجمل اللغوية الموجودة فى شكل ٤-١٧. المطلوب تمثيلها باستخدام المنطق الرياضى و الإجابة على السؤال التالى :

Is AlWalid alive now?

1. AlWalid was a fighter.
2. AlWalid was a human.
3. AlWalid born in 800 AD.
4. All fighters are champions.
5. All fighters at that time died in 850 AD.
6. No human lives more than 200.
7. It is now 2005.
8. If someone dies, then he is not alive at all later time.

شكل ٤-١٧ : جمل لغوية لوصف قصة.

الحل :

- تحويل الجمل اللغوية إلى التمثيل الرياضى المنطقى كما نرى فى الشكل ٤-١٨. نفترض أن t_1 هو تاريخ الميلاد و t_2 هو تاريخ اليوم كما فى الجملة رقم ٦. كذلك نفترض أن تاريخ الوفاة هو t_1 كما فى الجملة رقم ٨.

- إجابة السؤال بالنفى فتم تحويل الإجابة إلى التمثيل الرياضى المنطقى كما يلى :

Is AlWalid alive now?

No AlWalid is not alive now.

$\neg \text{alive}(\text{AlWalid}, \text{now}).$

1. fighter (AlWalid). 2.human (AlWalid).
3. born (AlWalid, 800).
4. $\forall x: \text{fighter}(X) \rightarrow \text{champion}(X)$.
5. $\forall x: \text{fighter}(X) \rightarrow \text{died}(X, 850)$.
6. $\forall x: \forall t_1: \forall t_2: \text{human}(X) \wedge \text{born}(X, t_1) \wedge \text{gt}(t_2 - t_1, 200) \rightarrow \neg \text{alive}(X, t_2)$.
7. now = 2005.
8. $\forall x: \forall t_1: \forall t_2: \text{died}(X, t_1) \wedge \text{gt}(t_2, t_1) \rightarrow \neg \text{alive}(X, t_2)$.

شكل ٤-١٨ : التمثيل الرياضى المنطقى للجمل اللغوية

الموجودة فى شكل ٤-١٧.

- استخدام التسلسل العكسى (لإثبات صحة الفرض أو عدمه). نجد فى الجمل الموجودة فى شكل ٤-١٧ أن هناك الطرف الأيمن فى كل من القاعدتين ٦ و ٨ مثل الفرض. لذلك نجرب الحل بطريقتين كما فى الشكلين ٤-١٩ و ٤-٢٠.

$\neg \text{alive}(\text{AlWalid}, \text{now})$

↑ من ٦ وبالتعويض $X=\text{AlWalid}, t_2=\text{now}$

$\text{human}(\text{AlWalid}) \wedge \text{born}(\text{AlWalid}, t_1) \wedge \text{gt}(\text{now} - t_1, 200)$

↑ من ٢

$\text{born}(\text{AlWalid}, t_1) \wedge \text{gt}(\text{now} - t_1, 200)$

↑ من ٣ وبالتعويض $t_1=800$

$\text{gt}(\text{now} - 800, 200)$

↑ من ٧

$\text{gt}(2005 - 800, 200)$

↑

nil

شكل ٤-١٩ : الحل باستخدام التسلسل العكسى مع اختيار قاعدة ٦.

→ alive(AlWalid, now)

↑ من ٩ وبالتعويض $X=AlWalid, t2=now$

$died(AlWalid, t1) \wedge gt(now-t1)$

↑ من ٥ وبالتعويض $X=AlWalid, t1=800$

$fighter(AlWalid) \wedge gt(now,800)$

↑ من ١

$gt(now,800)$

↑ من ٧

$gt(2005,800)$

↑

nil

شكل ٤-٢٠ : الحل باستخدام التسلسل العكسي مع اختيار قاعدة ٨.

كما رأينا أن الحلين الموجودين في الشكلين السابقين قادونا إلى نفس النتيجة و هي أن الفرض صحيح و الإجابة لا.

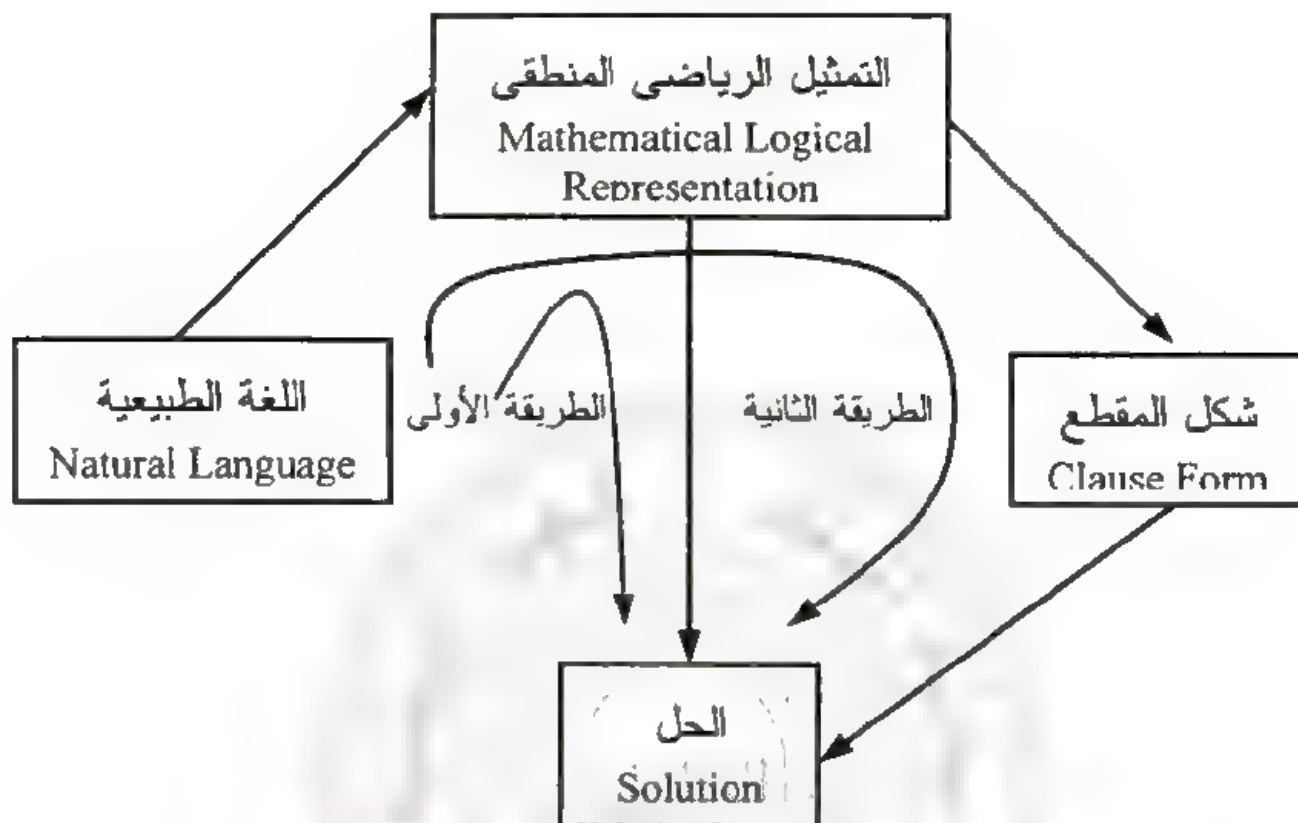
٤-٣ الحل باستخدام التوحيد/المطابقة و الحل

Solving using Unification and Resolution

فـيـمـا سـبـق كـنا نـقـوم بـتـحـوـيـل الجـمـل مـن اللـغـة الطـبـيـعـيـة (Natural Language) إـلـى لـغـة المنـطـق الـريـاضـي (Mathematical Logical Language)، ثم نستخدم ذلك التمثيل الرياضي المنطقي في الحصول على الحل.

نستطيع استخدام طريقة مختلفة للوصول إلى حل أو استنتاج حقائق جديدة و ذلك بتحويل الجمل المُمَثَّلة في التمثيل الرياضي المنطقي إلى شكل جديد هو شكل

المقطع (clause form) واستخدام تقنية الاتحاد و قواعد الحل (resolution) فى الوصول إلى الحل. شكل ٢١-٤ يعرض رسماً تخطيطياً يوضح طريقتى الحل.



شكل ٢١-٤ : رسم تخطيطى لطريقتى الحل
باستخدام التمثيل الرياضى المنطقى.

١-٣-٤ تحويل النموذج المنطقى إلى شكل المقطع

Mapping between Logical Model To Clause Form

عملية التحويل من التمثيل الرياضى المنطقى (Mathematical Logical Representation) إلى شكل المقطع (Clause Form) تخضع لقواعد عامة موجودة فى الجدول ٣-٤. يتضمن شكل المقطع الحقائق و القواعد كما سبق وأسلفنا فى التمثيل الرياضى.

جدول ٤-٣ : قواعد التحويل من التمثيل الرياضى إلى شكل المقطع

رقم	التمثيل الرياضى Mathematical Logic	شكل المقطع Clause Form
1	$R \rightarrow L$	$\neg R \vee L$
2	$\neg(\neg M)$	M
3	$\neg(M \wedge N)$	$\neg M \vee \neg N$
4	$\neg(M \vee N)$	$\neg M \wedge \neg N$
5	$M \vee (N \vee O)$	$(M \vee N) \vee O$
6	$(M \wedge N) \vee O$	$(M \vee O) \wedge (N \vee O)$

■ قواعد التحويل

نقرأ الآن جدول ٤-٣ الذى يحتوى على ٦ قوانين نفسرهم كما يلى :

- القانون رقم ١ أهم القوانين و يعنى أن أى قاعدة تتكون من طرفان، الطرف الأيسر (L) و الطرف الأيمن (R) - بصرف النظر عن عدد المسندات الموجودة فى كل طرف- تتحول إلى شكل المقطع بنفى الطرف الأيسر (بكافة ما يشمل من مسندات) مع ربطة بعلاقة OR مع الطرف الأيمن (بكافة ما يشمل من مسندات). ثم نطبق أياً من القوانين الأخرى حسب الحاجة.
- القانون رقم ٢ يعنى أن نفي النفي اثبات.
- القانون رقم ٣ يعنى أن نفي مُجْمَل مُسندَيْن بينهما علاقة OR يتحول إلى نفي كل مُسند على حدة و بينهما علاقة AND.
- القانون رقم ٤ يعنى أن نفي مُجْمَل مُسندَيْن بينهما علاقة AND يتحول إلى نفي كل مُسند على حدة و بينهما علاقة OR.

- القانون رقم ٥ يعنى أن ربط مُسند بعلاقة OR مع مجمل مسندين بينهما علاقة OR يتحول إلى أن ترتبط المسندات الثلاثة بعلاقة OR بين كل منهما.
- القانون رقم ٦ يعنى أن ربط مُسند بعلاقة OR مع مجمل مسندين بينهما علاقة AND يتحول إلى ربط المُسند الأول بعلاقة OR مع كل من المسندين فى جزأين يربط بينهما العلاقة AND.

■ تحويل الحقيقة (Fact)

تأخذ الحقيقة فى شكل المقطع نفس الشكل فى التمثيل الرياضى دون أى تغيير يُذكر.

■ تحويل القاعدة (Rule)

أما القاعدة فتخضع فى تحويلها من التمثيل الرياضى المنطقى إلى شكل المقطع للقواعد الموجودة فى الجدول ٤-٣.

٢-٣-٤ المطابقة/التوحيد Unification

تقنية المطابقة/التوحيد (Unification Technique) هو عملية إجراء مناظرة أو مطابقة بين صيغتين (مُسندَيْن) لاكتشاف ما إذا كان ممكناً إجراء بعض التعويضات لتصبح الصيغتين متطابقتين. تخضع هذه العملية للقواعد التالية :

- اسمى المسندين متطابقين (إذا كانا غير متطابقين نفشل عملية المقارنة).
- عدد و ترتيب و نوع العناصر بداخل الأقواس متطابق.
- نقارن كل عنصر فى الصيغة الأولى بالعنصر المقابل فى الصيغة الثانية، و توجد ثلاثة احتمالات :

- أن يكون العنصران المتقابلان عبارة عن قيمتان ثابتتان، فإذا كانتا متساويتين يكون العنصران متطابقين.
- أن يكون العنصران المتقابلان أحدهما متغير والآخر قيمة ثابتة، فإذا أمكن التعويض عن المتغير بالقيمة الثابتة يكون العنصران متطابقين.
- أن يكون العنصران المتقابلان متغيران، فيمكن التعويض عن (احلال) أحدهما محل الآخر ليصبحا متطابقين.

مثال ٥ : هل الصيغتين التاليتين متطابقتين :

$$M(x, x), M(y, z)$$

الحل : لكى تصبح الصيغتان متطابقتين يجب أن نطبق قواعد المقارنة السابقة كما يلى :

أولاً : اسما الصيغتين متطابقتان.

ثانياً : عدد و ترتيب و نوع العناصر داخل الأقواس متطابق.

ثالثاً : نقارن العناصر المتقابلة فى الصيغتين فنجد أن الجميع متغيرات فنقوم بالآتى:

- التعويض عن x بـ y أى نُحل y محل x (كما فى ناحية اليسار). ثم نعوض عن y بـ z أو z بـ y .
- أو التعويض عن y بـ x أى نُحل x محل y (كما فى ناحية اليمين). ثم نعوض عن x بـ z أو z بـ x .

$$1) x=y :$$

$$M(y,y), M(y,z)$$

$$2) y=z :$$

$$M(z,z),$$

$$M(z,z)$$

$$2) z=y :$$

$$M(y,y),$$

$$M(y,y)$$

$$1) y=x :$$

$$M(x,x), M(z,z)$$

$$2) x=z :$$

$$M(z,z),$$

$$M(z,z)$$

$$2) z=x :$$

$$M(x,x),$$

$$M(x,x)$$

مما سبق يتضح أنه يمكن أن تكون الصيغتان $M(y,z)$ و $M(x,x)$ متطابقتين في أربع حالات كما هو مبين في السطور السابقة.

مثال ٦ : هل الصيغتين التاليتين متطابقتين :

$eats(x, mango)$, $eats(y,z)$

الحل : لكي تصبح الصيغتان متطابقتين يجب أن نطبق قواعد المقارنة السابقة كما يلي :

أولاً : اسما الصيغتين متطابقتان.

ثانياً : عدد و ترتيب و نوع العناصر داخل الأقواس متطابق.

ثالثاً : نقارن العناصر المتقابلة في الصيغتين فنجد أن العنصر الأول في الصيغتين متغير و العنصر الثانى ثابت فى الصيغة الأولى و متغير فى الصيغة الثانية، فنقوم بالآتى:

- التعويض عن x بـ y أى نُحِلْ y محل x (كما فى ناحية اليسار).
- أو التعويض عن y بـ x أى نُحِلْ x محل y (كما فى ناحية اليمين).
- بعد ذلك فى الحالتين لابد من التعويض عن ثم نعوّض عن z بـ $mango$ أى نستبدل x بـ $mango$.

1) $x=y$:		1) $y=x$:
$eats(y, mango)$, $eats(y,z)$		$eats(x, mango)$, $eats(x,z)$
2) $z=mango$		

$eats(y,mango),eats(y,mango)$ | $eats(x,mango),eats(x,mango)$

مما سبق يتضح أنه يمكن أن تكون الصيغتان $eats(x, mango)$ و $eats(y, z)$ متطابقتين فى حالتين فقط كما هو مبيّن فى السطور السابقة.

مثال ٧ : هل الصيغتين التاليتين فى كل حالة متطابقتين :

- 1) $eats(x, mango), likes(x, mango)$
- 2) $eats(x, mango), eats(y, Orange)$
- 3) $eats(x, mango), eats(y, mango, z)$

الحل :

- (١) الصيغتان غير متطابقتين لاختلاف اسم المُسند فى الصيغتين.
- (٢) الصيغتان غير متطابقتين لاختلاف القيم الثابتة للعنصر الثانى فى المسند الأول عن نظيرتها المقابلة لها فى الصيغة الثانية.
- (٣) الصيغتان غير متطابقتين لاختلاف عدد العناصر بين الأقواس.

تنبيه هام :

- يجب أن يتم التعويض من اليسار إلى اليمين.
- الهدف من تقنية الاتحاد هو محاولة إيجاد تعويض واحد على الأقل يجعل الصيغتين متطابقتين.

Resolution

٤-٣-٣ الحل

الحل (Resolution) هو الطريقة التى تعمل على الجمل التى حوّلت إلى شكل قياسى مريح عن طريق تفنيد هذه الجمل باستخدام التسلسل العكسى (backward chaining). و يتم ذلك باستخدام القاعدتين التاليتين :

- إذا احتوت جملتان على مسندين T و $\neg T$ أى أن المُسندين متطابقين و أحدهما منفى و الآخر مُثبت تكون الجملتان مكملتان complement و

يتم حذف المسند ونفيه من الجملتين ويتبقى لدينا المسندات الأخرى من الجملتين فنجمعهم فى جملة واحدة و بينهما OR.

■ إذا كان هناك جملتان هما $M \vee N$ و N فإن $M \vee N$ تؤول إلى M .

٤-٣-٤ التمثيل و الاستنتاج باستخدام التوحيد و الحل

Representation & deduction by Unification & Resolution

فى السابق استطعنا الحصول على إجابة على بعض الأسئلة مباشرة من التمثيل الرياضى المنطقى (mathematical logical representation) لمجموعة جمل لغوية كما فى الطريقة الأولى فى شكل ٤-١٧.

أما الآن فيمكننا استخدام الطريقة الجديدة المتمثلة فى شكل ٤-٢١. هذه الطريقة تتضمن تحويل الجمل اللغوية إلى تمثيل رياضى منطقى أولاً، ثم تحويل التمثيل الرياضى المنطقى باستخدام القوانين الموجودة فى جدول ٤-٣ إلى شكل المقطع (clause form) ثم نستخدم قواعد Resolution مع تقنية المطابقة و التوحيد للوصول إلى الحل.

الجدير بالذكر أنه عند الوصول فى الحل إلى nil (لاشئ) فى هذه الطريقة، فإن ذلك يعنى أن الفرض خطأ و هذا عكس الموقف فى الطريقة الأولى عند الحل باستخدام التمثيل الرياضى المنطقى فقط.

مثال ٨

بالعودة إلى الجمل اللغوية الموجودة فى مثال ١ فى شكل ٤-٨ قم بالآتى :

أ- حول الجمل الإنجليزية إلى التمثيل الرياضى.

ب- أحب على السؤال : does Alla study editing_report?

ت- حول التمثيل الناتج من الفقرة (أ) إلى clause form.

ث- باستخدام التمثيل الناتج و قواعد resolution و تقنية الاتحاد أجب على

السؤال : does Alla study editing_report?

الحل :

أ و ب : سبق و حولنا الجمل اللغوية فى شكل ٨-٤ إلى التمثيل الرياضى الموجود فى شكل ٩-٤ و استخدناه فى الإجابة على نفس السؤال فى شكل ١٠-٤ و ١١-٤ و كانت الإجابة نعم. شكل ٢٢-٤ يعرض الجمل اللغوية و يلى كل جملة لغوية التمثيل الرياضى المنطقى ((Mathematical Logic(ML)).

ت. المطلوب الآن هو تحويل الجمل المُمَثَّلة بالمنطق الرياضى إلى شكل المقطع باستخدام القوانين المناسبة. شكل ٢٢-٤ يضم أيضاً شكل المقطع (Clause Form (CF)) حيث يلى كل جملة لغوية و تمثيلها الرياضى المنطقى.

1. Database is a computer course.

ML: computer_course (database).

CF : computer_course (database).

2. Editing reports is a language course.

ML: language_course (editing_reports).

CF : language_course (editing_reports).

3. Ahmad likes difficult courses and easy courses.

ML:

$\forall x: \text{difficult_course}(X1) \vee \text{easy_course}(X1) \rightarrow \text{studies}(\text{Ahmad}, X1)$

CF :

$\neg [\text{difficult_course}(X2) \vee \text{easy_course}(X2)] \vee \text{studies}(\text{Ahmad}, X2).$

$[\neg \text{difficult_course}(X2) \wedge \neg \text{easy_course}(X2)] \vee \text{studies}(\text{Ahmad}, X2).$

$(\neg \text{difficult_course}(X2) \vee \text{studies}(\text{Ahmad}, X2)) \wedge$

$(\neg \text{easy_course}(X2) \vee \text{studies}(\text{Ahmad}, X2)).$

4. Artificial Intelligence is a computer course.

ML: $\text{computer_course}(\text{artificial_intelligence})$.

CF : $\text{computer_course}(\text{artificial_intelligence})$.

5. Sayf study any difficult course.

ML: $\forall x: \text{difficult_course}(X) \rightarrow \text{studies}(\text{Sayf}, X)$.

CF : $\neg \text{difficult_course}(X3) \vee \text{studies}(\text{Sayf}, X3)$.

6. Computer courses are not easy course.

ML: $\forall x: \text{computer_course}(X) \rightarrow \neg \text{easy_course}(X)$.

CF : $\neg \text{computer_course}(X4) \vee \neg \text{easy_course}(X4)$.

7. Language courses are difficult.

ML: $\forall x: \text{language_course}(X) \rightarrow \text{difficult_course}(X)$.

CF : $\neg \text{language_course}(X5) \vee \text{difficult_course}(X5)$.

8. Ahmad a computer student.

ML: $\text{computer_student}(\text{Ahmad})$.

CF : $\text{computer_student}(\text{Ahmad})$.

9. Sayf is a computer student.

ML : $\text{computer_student}(\text{Sayf})$.

CF : $\text{computer_student}(\text{Ahmad})$.

10. Alla studies whatever Sayf studies.

ML: $\forall x: \text{studies}(\text{Sayf}, X) \rightarrow \text{studies}(\text{Alla}, X)$.

CF : $\neg \text{studies}(\text{Sayf}, X6) \vee \text{studies}(\text{Alla}, X6)$.

شكل ٤-٢٢ : جمل فى التمثيل اللغوى و الرياضى المنطقى و شكل المقطع.

ث. الآن جاء دور تقنية التوحيد و قواعد Resolution للإجابة على السؤال :

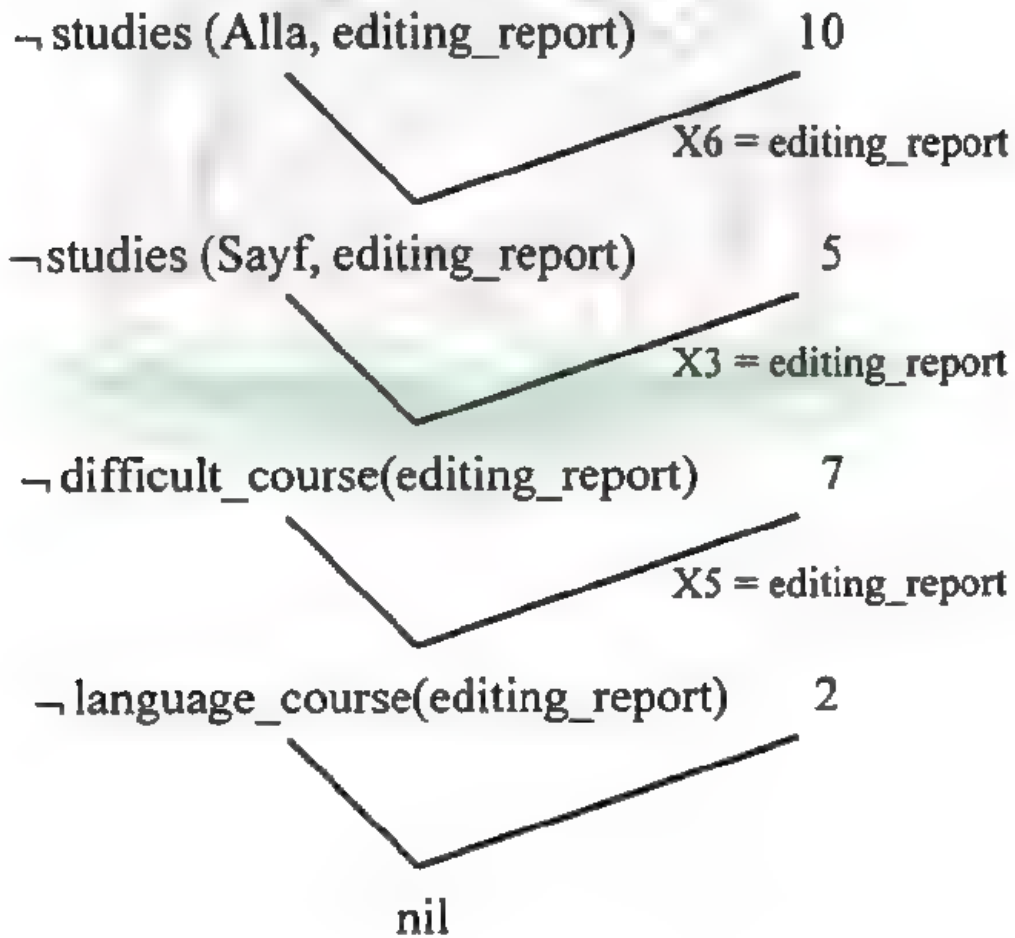
does Alla study editing_report?

نحول السؤال إلى التمثيل المنطقى بعد الإجابة عليه بالنفى "لا" كما يلي

No Alla doesn't study editing_report.

\neg studies (Alla, editing_report)

ثم نستخدم هذه الصيغة مع الصيغ فى شكل المقطع مع قواعد resolution و تقنية التوحيد كما فى شكل ٤-٢٣.



شكل ٤-٢٣ : الحل باستخدام شكل المقطع و تقنية التوحيد و قواعد Resolution.

بدأنا بالبحث عن صيغة من هذه الصيغ يمكن تطبيق قواعد resolution و تقنية التوحيد عليها (صيغة تحتوى على مسند متطابق مع تمثيل السؤال و مُثَبَّة لتطبيق القاعدة الأولى أو صيغة متطابقة و منفية مثلها لتطبيق القاعدة الثانية).

عند الوصول إلى nil (لا شىء) فى هذه الطريقة، فإن ذلك يعنى أن الفرض خطأ و العكس صحيح أى أن الإجابة هى نعم آلاء تدرس تحرير التقارير مثل الإجابة التى حصلنا عليها فى المثال الأول باستخدام طريقة الحل الأولى.

٤-٤ التمثيل المنطقى لشبكة المعرفة اللفظية

Logical Representation of Semantic Net

سبق أن تعرّفنا فى الفصل الثانى على شبكة المعرفة اللفظية و علمنا أن الفكرة الرئيسية وراء شبكة المعرفة اللفظية تكمن فى أن معنى المفهوم يأتى من طرق ربطه أو اتصاله بالمفاهيم الأخرى. علمنا أيضاً أن شبكة المعرفة اللفظية عبارة عن مجموعة من العقد أو النقاط (nodes) متصلة ببعضها البعض بواسطة مجموعة من الأقواس أو الأسهم التى تحمل عنواناً لتمثل العلاقات الرابطة بين العقد.

٤-٤-١ طريقة التمثيل Representation Approach

تمثيل شبكة المعرفة باستخدام لغة المنطق الرياضى يجب أن يتضمن تمثيل كلاً من : جميع الفئات/الطبقات (classes) و العلاقات الرابطة بينها (isa) و الكائنات/الحالات (objects) و العلاقات التى تربطها بالطبقة المنتمية إليها (instance) و خصائص كل طبقة و كل كائن (attributes). يتم التمثيل طبقاً للأسلوب التالى :

- أى صفة لطبقة يتم تمثيلها بقاعدة (إمكانية تعميمها على كل كائن ينتمى إلى تلك الطبقة).

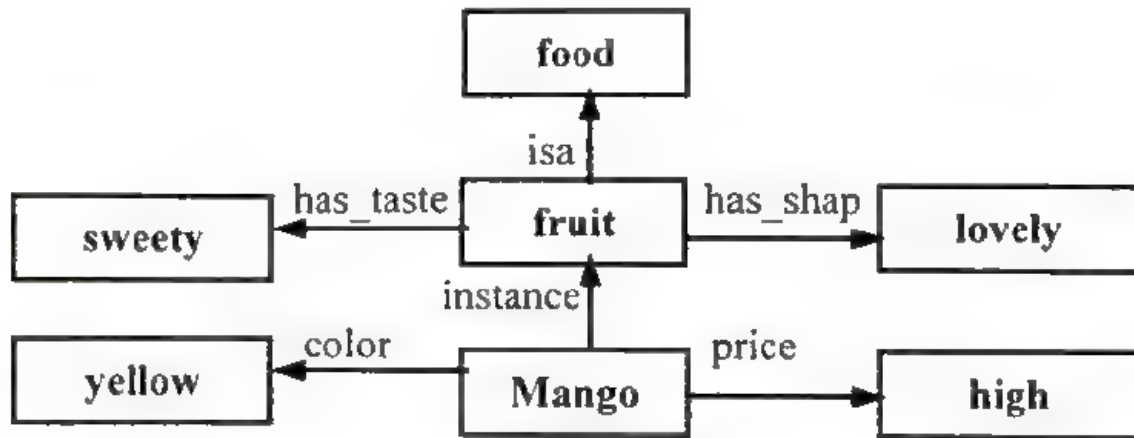
- أى صفة لكائن يتم تمثيل بحقيقة (لكونها خاصية معلوم قيمتها لكائن محدد و لا يمكن تعميمها على أى كائن آخر).
- فى حالة وجود صفتين أو أكثر لطبقة يتم تمثيلهما فى قاعدة واحدة أو تمثيل كلاً منهما فى قاعدة منفصلة.
- العلاقة الرابطة isa يمكن تمثيلها بالطريقتين التاليتين :
 - قاعدة تدل على أن أى كائن ينتمى إلى الطبقة الابن ينتمى بالضرورة إلى الطبقة الأب أيضاً.
 - حقيقة (مُسند) اسمها هو العلاقة الرابطة isa و اسم الطبقة الابن و الطبقة الأب هما العنصرين بين القوسين.
- العلاقة الرابطة instance يمكن تمثيلها بالطريقتين التاليتين :
 - حقيقة (مُسند واحد) اسمها هو اسم الطبقة و اسم الكائن هو العنصر بين القوسين.
 - حقيقة (مُسند واحد) اسمها هو العلاقة الرابطة instance و اسم الكائن و الطبقة التى ينتمى إليها هما العنصرين بين القوسين.

٤-٤-٢ مثال ٨

سبق أن عرَضْنَا فى الفصل الثانى شكل ٢-٩ و نعيده الآن فى شكل ٤-٢٤ كمثال لشبكة معرفة لفظية. يعرض شكل ٤-٢٤ شبكة معرفة لفظية تتضمن طبقة الفاكهة fruit ترتبط بطبقة أعلى هى طبقة الطعام food من خلال العلاقة الرابطة isa. تحتوى الشبكة أيضاً على كائن المانجو mango المنتمى إلى طبقة الفاكهة عن طريق العلاقة الرابطة instance.

نرى أيضاً الخاصيتين has_taste (له مذاق) لطبقة الفاكهة و قيمتها sweet (حلوة المذاق) و has_shape (له شكل) و قيمتها lovely (فائقة). و

الخاصيتين color (اللون) و قيمتها yellow (أصفر) و price (السعر) و قيمتها expensive (غالية).



شكل ٤ : شبكة معرفة لفظية لطبقات منتجات استهلاكية.

الحل : يمكن تمثيل شبكة المعرفة اللفظية الموجودة في شكل ٤-٢٤ كما يلي :

■ التمثيل بلغة المنطق الرياضى باستخدام isa و instance كما في الشكل ٤-٢٥.

1. isa(fruit, food).
2. instance(mango, fruit).
3. $\forall x: \text{instance}(X, \text{fruit}) \rightarrow \text{has_taste}(X, \text{sweety}) \wedge \text{has_shape}(X, \text{lovely}).$
4. price(mango, high).
5. color(mango, yellow).

شكل ٤-٢٥ : تمثيل شبكة المعرفة باستخدام isa و instance

■ التمثيل بلغة المنطق الرياضى بدون isa و instance كما في الشكل ٤-

1. $\forall x: \text{fruit}(X) \rightarrow \text{food}(X).$
2. $\text{fruit}(\text{mango}).$
3. $\forall x: \text{fruit}(X) \rightarrow \text{has_taste}(X, \text{sweet}).$
4. $\forall x: \text{fruit}(X) \rightarrow \text{has_shape}(X, \text{lovely}).$
5. $\text{price}(\text{mango}, \text{high}).$
6. $\text{color}(\text{mango}, \text{yellow}).$

شكل ٤-٢٦ : تمثيل شبكة المعرفة بدون isa و instance.

٤-٥ أسئلة

١. عرف Unification Technique و Resolution

٢. اختبر تطابق الجمل التالية :

love (x, y), love (z, computer)
 love (x, y), love (computer)
 love (Ali, English), hate (Ali, English)
 love (Ali, English), love (Walid, English)

٣. فى مثال (١)، أجب على السؤال التالى :

Does Alla study Database?

يمكنك إضافة جمل لغوية جديدة إذا استدعى الأمر ذلك (اقتدى بما فعلناه فى مثال ٣).

٤. فى مثال ٢، حول التمثيل الرياضى الموجود فى شكل ٤-١٣ إلى شكل المقطع ثم استخدم قواعد Resolution و تقنية التوحيد للإجابة على السؤال التالى :

Does Amr likes mango?

٥. حول الجمل التالية إلى Predicate Logic :

1. Lolo only likes easy courses.
2. Math courses are hard.
3. All courses in the department plan are easy.
4. csc363 is a course in the department plan.
5. csc425 is a course in the department plan.
6. people who live in Saudi Arabian are Saudi or foreign.

ثم أجب على السؤالين التاليين :

Does lolo likes csc426?

Does lolo likes csc325?

٦. حول الجمل التالية إلى Predicate Logic :

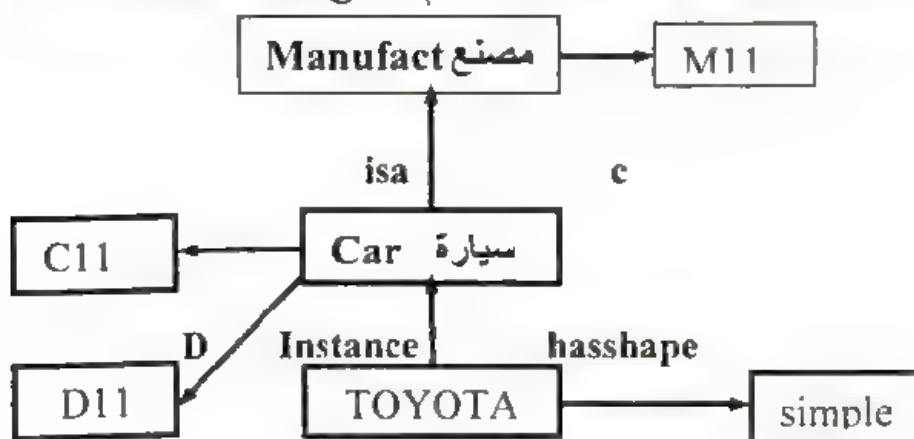
7. Tuna is a cat.
8. Nada owns a horse.
9. every horse owner is an animal lover.
10. No animal lover kills an animal.
11. Either Ali or Nada killed the cat.
12. A cat is an animal.

ثم أجب على السؤالين التاليين :

Did Ali kill tuna?

Did Nada kill tuna?

٧. مثل Semantic Net التالية باستخدام Predicated Logic :-



٨. حول الجمل الانجليزية التالية إلى التمثيل الرياضى المنطقى فقط :

1. Omar was Arabian.
2. Omar was born in 600 AD.
3. All men are mortal.
4. No mortal lives more than 150.
5. It is now 2002.
6. Alive means dead.
7. If some one die, then he is dead at all later times.

٩. باستخدام التمثيل الرياضى المنطقى التالى :

1. fruit (Apple).
2. human (Fahd).
3. $\forall x: \text{fruit}(x) \rightarrow \text{food}(x)$.
4. $\forall x: \forall y: \text{fruit}(x) \wedge \text{fresh}(x) \wedge \text{human}(y) \rightarrow \text{eat}(y,x)$.
5. $\forall x: \text{eat}(\text{Fahd},x) \rightarrow \text{eat}(\text{Lolo},x)$
6. $\forall x: \forall y: \text{human}(y) \wedge \text{food}(x) \rightarrow \text{eat}(y,x)$

أجب على السؤال التالى :

Does Lolo eats mango?

ثم حول التمثيل المنطقى إلى Clause Form

و أجب على نفس السؤال باستخدام unification و clause form.



الفصل الخامس

معالجة اللغات الطبيعية

Natural Language Processing (NLP)

يعتقد الكثير من المتخصصين في الذكاء الاصطناعي أن أحد المهام الهامة التي يمكن أن ينجح الذكاء الاصطناعي في إنجازها هو معالجة اللغات الطبيعية. اللغات الطبيعية هي اللغات الحية التي يتحدث بها الإنسان سواء كانت عربية أو إنجليزية أو فرنسية أو صينية أو غيرهم من اللغات. و عند إنجاز تلك المهمة ستفتح معالجة اللغات الطبيعية باباً للحوار المباشر بين الحاسب و الإنسان، الذي يمكن أن يتجاوز البرمجة العادية و نظام التشغيل. في هذا الفصل نركز على الإعراب و في الفصل التالي نقدم تطبيقاً لفهم اللغة العربية و تمثيل المعرفة منطقياً باستخدام لغة PROLOG .

Introduction

١-٥ مقدمة

نشأت حقول علمية جديدة تحت مظلة الذكاء الاصطناعي كان من أهمها حقل معالجة اللغات الطبيعية. تطور هذا الحقل ليشمل العديد من جوانب معالجة اللغات الطبيعية من نواحي الكتابة وتصحيح الإملاء والقواعد النحوية والصرف والمعاني والترجمة و البلاغة والشعر.

كما و أن اللغويين مهتمون باللغات الطبيعية فإنهم مهتمون أيضاً باستخدام الحاسب الآلى فى مجال اللغويات. ولقد أنجز باحثو الذكاء الاصطناعى العديد من الأبحاث و حققوا إنجازات باهرة على مدار السنوات الأربعون الماضية. لكن نظراً لحجم و صعوبة اللغات الإنسانية فلم تكتمل معالجتها بالكامل. و مع أن الباحثون قد طوروا البرامج التى تفهم معنى الجمل فلا زالت هناك صعوبات أمام الوصول إلى ذلك الهدف.

لقد نشأت بعض التقنيات المساعدة فى معالجة اللغات الطبيعية. و منذ عشرون عاماً تقريباً بدأ الاهتمام بمعالجة اللغة العربية و فهم الحاسب لها تضمنت دراسة قواعد النحو العربى و نماذج الصرف المختلفة و طرق اشتقاق جذور الكلمات العربية. و لقد كانت هناك بحوث شيقة و قيّمة فى مجال فهم العديد من اللغات الطبيعية المتعددة. الفصل التالى يعرض تطبيقاً كاملاً لفهم اللغة العربية.

الهدف الرئيسى لمعالجة اللغات الطبيعية (Natural Language Understanding (NLP)) هو جعل الحاسب قادراً على فهم الأوامر المكتوبة بلغات البشر القياسية. وظيفة مُعالج اللغة الطبيعية (Natural Language Processor) هى استخلاص المعلومات من الجمل المُدخلة عن طريق وحدة المفاتيح أو من ملف (أى أنه لا يهتم بطريقة دخول الجمل إلى الحاسب الآلى).

يمكن أن توفر NLP واجهة اتصال لبرامج الحاسب و خصوصاً لبرامج إدارة قواعد البيانات و برامج حل المشاكل. و هى ضرورية فى مجال الترجمة من لغة طبيعية إلى لغة طبيعية أخرى. مجال NLP مجال واسع جداً و توجد طريقتين لمعالجة اللغة الطبيعية :

▪ الطريقة الأولى تحاول استخدام كافة المعلومات الموجودة في الجملة المُدخَّلة، تماماً مثل ما يحدث في الواقع الإنساني. هدف هذه الطريقة هو جعل الحاسب الآلي قادراً على الدخول في محادثة. و هذا صعب تماماً أن يتحقق بشكل كامل.

▪ الطريقة الثانية تحاول جعل الحاسب الآلي يقبل أوامر أو جمل باللغة الطبيعية و يستخلص منها المعلومات الأساسية الموجودة في المُدخَّلات.

فهم اللغات الطبيعية يتطلب حقائق عن ترتيب الكلمات في الجمل (شجرة الإعراب) وعن المعاني وتظهر في نماذج التمثيل للمجالات المختلفة و عن طريقة اشتقاق الكلمات من المصادر أو الجذور .

نستطيع القول أن فهم اللغات الطبيعية أساسه التمثيل الجيد، فالتمثيل الجيد هو مفتاح النجاح. و نظراً لوجود فجوة بين البيانات وفهم البيانات فإن التمثيل الجيد هو الجسر الذي يعبر هذه الفجوة.

٥-٢ اللغات و النحو Language and Grammar

التكثيف مع صعوبة و مرونة اللغات البشرية هو أحد السمات الأكثر صعوبة في بناء نظام لمعالجة اللغة الطبيعية. أثناء بناء نظام معالجة اللغات الطبيعية، من المهم تقييد أنواع الجمل التي يُتَوَقَّع أن يفهما المعالج كجزء من أحد اللغات الطبيعية. يمكن تحديد أى لغة طبيعية باستخدام قواعد تُشكِّل النحو الذي يتحكم باستخدام تلك اللغة.

Grammar Description

٥-٢-١ وصف النحو

الخطوة الأولى في بناء نظام NLP هي وصف النحو الذي يتضمن التراكيب التي سيدركها ذلك النظام. استخدام النحو له عدة مميزات هي :

- يُعطى فهماً دقيقاً و سهلاً للتوصيف النحوي للجمل المُدخلة من لغة ما.
- يمكن إنشاء برنامج إعراب الجمل (parser) للنحو المُصنَّم جيداً.
- يحدد النحو تراكيب اللغة المُدخلة.

لذلك يقوم النظام بترجمة الجمل المُدخلة إلى الترميز الداخلي (object code) ببساطة و يكتشف أى أخطاء لغوية تخالف القواعد و التراكيب المُحددة في النحو المُستخدم. و يمكن تحديد النحو بطرقٍ مختلفة. واحدة من أفضل الطرق لتحديد مواصفات النحو هي طريقة شكل BNF (Backus Naur Form).

بصفة عامة يتضمن النحو أربع كيانات و هي :

- الجزء النهائي (Terminal) هي الرموز الأساسية التي تتكون منها الجمل في أى لغة طبيعية. الكلمة لفظة (token) مرادفة للجزء النهائي (Terminal).

- الجزء اللانهائي (Non-Terminal) هي رموز خاصة يمكن إحاطتها بواسطة <...> و ترمز إلى مجموعة من السلاسل الحرفية (الكلمات). تأمل القاعدة التالية :

<SENTENCE> ::= <SUBJECT> likes <OBJECT>

<مفعول به> likes <فاعل> ::= <جملة>

الأجزاء اللانهائية هي <SENTENCE> و <SUBJECT> و <OBJECT>. بينما likes هي جزء نهائى.

▪ رمز البداية (Start Symbol) هو جزء لا نهائى يبدأ عنده المعالج عند حساب كيفية معالجة الجملة المُدخلة مثل <SENTENCE> فى القاعدة السابقة.

▪ قواعد الإنتاج (Production Rules) و هي تُعرّف الطرق التى يمكن بها بناء تراكيب نحوية من تراكيب أخرى و من الأجزاء اللانهائية. القاعدة السابقة شكل من أشكال قواعد الإنتاج.

نلتزم ببعض التقاليد عند استخدام الأجزاء اللانهائية فى قواعد BNF و هي:

- توضع الأجزاء اللانهائية بين علامتين "<....>" مثل <Noun> على سبيل المثال.
- تُستخدَم النجمة "*" فى قواعد الإنتاج للإشارة إلى عنصر قادر على تمثيل جزء لا نهائى.
- تُستخدَم علامة الجمع "+" فى قواعد الإنتاج للإشارة إلى إمكانية وجود مثال/حالة أو أكثر.
- تُستخدَم العلامة "|" فى قواعد الإنتاج للإشارة إلى العلاقة المنطقية OR (أو).
- الجزء اللانهائى المُحاط بالقوسين المربعين "[...]" فى قاعدة إنتاج يكون جزء اختياري غير مُلزم.

Formal Grammar

٥-٢-٢ النحو المُشكَّل

النحو المُشكَّل هو مجموعة -ممکن لا نهائية- من سلاسل حرفية (strings) ذات طول مُحدَّد مُشكَّلة من مجموعة محددة من المفردات اللغوية. و هناك أربع أنواع من النحو هي :

- **نحو من النوع صفر (Type 0 Grammar)** و يُعرَّف على أنه مجموعة من قواعد الإنتاج مُشكَّلة من رموز مفردات لغوية دون قيد على شكل القواعد. يشبه هذا آلة تيورنج (Turing Machine) التي تستجيب لجمل محددة يمكن توليدها من اللغة.
- **نحو من النوع ١ (Type 1 Grammar)** و يُعرَّف على أنه مثل النوع صفر إذا كان شكل القواعد مُقيَّد بحيث أنه لكل قاعدة $X \rightarrow Y$ من النحو، يكون الطرف الأيمن (Y) يحتوى على عدد من الرموز مساوٍ للرموز الموجودة في الطرف الأيسر (X). يُطلق على النوع ١ سياق النحو الحساس (Context-Sensitive Grammar). كمثال لهذا النحو افترض أن رمز البداية S و الأجزاء النهائية هي a و b و c يكون النحو كما يلي :

$$S \rightarrow aSBC, S \rightarrow aBC, CB \rightarrow BC, aB \rightarrow ab, \\ BB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc$$

اللغة التي تم توليدها بواسطة هذا النحو هي مجموعة السلاسل الحرفية abc و aabbcc و aaabbbcccc و هذه اللغة - حيث يجب أن يحدث كل رمز نفس العدد من المرات و يجب أن يظهر في المكان الملائم في السلسلة الحرفية- لا يمكن توليدها بواسطة أى نحو من نوع أكثر تقيُّداً.

■ **نحو من النوع ٢ (Type 2 Grammar)** و هو سياق النحو الحر **(Context-Free Grammar)**، حيث يجب أن تحتوى كل قاعدة إنتاج على رمز جزء لا نهائى واحد فقط فى الطرف الأيسر. على سبيل المثال، سياق النحو الحر الذى يؤكّد الجمل ab و $aabb$ و $aaabbb$ و ... هو :

$$S \rightarrow aSb, S \rightarrow ab$$

شكل ١-٥ يعرض مثالاً لسياق النحو الحر الذى يؤكّد بعض الجمل فى هذا المثال، الأجزاء النهائية هى the و a و $girl$ و boy و $work$ و eat و رمز البداية هو $\langle SENTENCE \rangle$.

$\langle SENTENCE \rangle \rightarrow \langle NOUN_PHRASE \rangle \langle VERB_PHRASE \rangle$
 $\langle NOUN_PHRASE \rangle \rightarrow \langle DETERMINER \rangle \langle NOUN \rangle$
 $\langle NOUN_PHRASE \rangle \rightarrow \langle VERB \rangle \langle NOUN_PHRASE \rangle$
 $\langle DETERMINER \rangle \rightarrow the, a, \dots$
 $\langle NOUN \rangle \rightarrow girl, boy, \dots$
 $\langle VERB \rangle \rightarrow work, eat, \dots$

شكل ١-٥ : مثال لسياق النحو الحر:

توجد ميزة هامة فى سياق النحو الحر ألا و هى أنه فى برامج معالجة اللغة الطبيعية يمكن تقليدياً تمثيل أى اشتقاق أو قاعدة فى شكل شجرة (Tree). غالباً ما يُطلق على سياق النحو الحساس و سياق النحو الحر اسم النحو المبنى على تركيب الجملة **(Phrase-Structured Grammar)**.

▪ **نحو من النوع ٣ (Type 3 Grammar)** حيث تأخذ كل قاعدة إنتاج الشكل :

$$X \rightarrow aY \text{ or } X \rightarrow a$$

حيث X و Y متغيرات مفردة و a جزء نهائى مفرد. هذا النحو يُمثل النحو النظامى. على سبيل المثال يمكن أن يُولد النحو النظامى مجموعة السلاسل الحرفية لعنصر a واحد أو أكثر متبوعاً بعنصر b واحد أو أكثر (مع عدم ضمان عدد متساوٍ من a 's أو b 's) كما يلي :

$$S \rightarrow aS, S \rightarrow aT, T \rightarrow b, T \rightarrow bT$$

٥-٣ تقنيات الإعراب Parsing Techniques

الإعراب (parsing) هو عملية تجزئ الجملة إلى كلمات مع معرفة إعراب كل كلمة وعلاقتها بباقي الكلمات وتخضع في ذلك لسياق قواعد النحو الحر (Context-Free Grammar).

نواة و قلب أى نظام لمعالجة لغة طبيعية هو المُعَرِّب (Parser). فالمُعَرِّب هو ذلك الجزء من برامج النظام الذى يقرأ كل جملة، كلمة تلو كلمة ليقرر الصالح من الطالح. فهو يستطيع أن يميز تركيب الجملة المُدخلة و يدقق كون كل لفظة جزء من نموذج صحيح مُحَدَّد من نحو اللغة الطبيعية، أى أن المُعَرِّب هو مُدَقِّق الأخطاء (error checking).

٥-٣-١ الإعراب من القمة لأسفل Top-down Parsing

يبدأ المُعَرِّب أولاً بالنظر إلى قواعد النحو بحثاً عن التركيبية (مقطع الجملة، أو غيرها) المطلوبة عن مستوى القمة. ثم يبحث عن قواعد لكل من مكونات تركيبية مستوى القمة تلك و يتقدم متجولاً حتى يُكْمِل بناء تركيبية كاملة للجملة المُدخلة من

مجموعة من القواعد. فإذا طبقت الجملة البيانات المُدخلة تنتهي عملية الإعراب بنجاح و إلا فإن برنامج المُعرب يعود لیبداً عن القمة مرة أخرى من تركيبة تالية ليولّد تركيبة أخرى للجملة.

یبدأ برنامج المُعرب بقاعدة الإنتاج الأكثر عمومية (رمز البداية) في النحو و يحاول أن يُنتج مجموعة من قواعد الإنتاج التي تُولّد الجملة. الإعراب من القمة لأسفل (top-down parsing) هو طريقة الإعراب المُعاود التنازلي (recursive descent parsing) للتحليل النحوي، حيث يتم تنفيذ مجموعة من الإجراءات (البرامج الصغيرة) لمعالجة النص المُدخّل. و يتم ربط إجراء بكل جزء لا نهائي من النحو.

٥-٣-٢ الإعراب من القاع لأعلى Bottom-up Parsing

یبدأ المُعرب أولاً بالنظر إلى قواعد النحو ليُوحّد كلمات الجملة المُدخلة في جزء من تركيبة أكبر (عبارات و مقاطع) و يستمر يحاول أن يوحّد أجزاء التركيبة ليبين كيف تُشكّل جميع كلمات الجملة المُدخلة لجملة صحيحة في النحو.

یبدأ برنامج المُعرب بالجملة المُدخلة و يعمل بطريقة عكسية ليثبت أنها حالة لأحد قواعد الإنتاج في النحو. و عادة ما يتطلب تناول جدول إعراب (parsing table). جدول الإعراب هو الجدول الذي يُسجّل أي قاعدة لنتاج يجب أن يستخدمها برنامج المُعرب مع رمز كل جزء نهائي.

عند العمل خلال الجملة المُدخلة، لن يكون المُعرب قادراً على أن يخبرنا - على سبيل المثال - هل الجملة التي بدأت بـ if ستكون في الشكل if .. then أو في الشكل if .. then .. else.

٥-٣-٣ الإعراب بإهمال الغير ضرورى Noise Disposal Parsing

هذا النوع من الإعراب يُستخدَم مع أي من الطريقتين السابقتين للإعراب مع التطبيقات التى تهتم بعدد قليل من الكلمات الأساسية التى تحتوى الجمل عليها و لا تهتم بكافة الكلمات الموجودة فى الجملة و تشكّل اللغة. فى الحقيقة، هذه الأنواع من التطبيقات تهتم فقط بالمعلومات التى تحتوى عليها الجملة أكثر من مكونات الجملة نفسها.

يتم ذلك بينما يعتبر برنامج المُعرِب كافة الكلمات المجهولة أو الغير مطلوبة ككلمات غير ضرورية (noise words) و تُستبعد. ببساطة يجب أن تلتزم جميع الجمل المُدخّلة بتشكيلات محددة أى قوالب صارمة (rigid format). هذه التشكيلات تعيد تجميع اللغة الطبيعية.

من عيوب طريقة الإعراب بإهمال الغير ضرورى هو أنها غير مفيدة خارج نطاق تطبيقات معينة مثل تطبيقات قواعد البيانات لأنها مبنية على فرضين. الفرض الأول هو أن الجملة تتبع قالب محدد بدقة. الفرض الثانى هو أن كلمات أساسية قليلة تُعد هامة (على الرغم من أنه فى المحادثة الفعلية تكون أغلب الكلمات هامة بطريقة أو بأخرى. العيب الثانى فى الإعراب بإهمال الغير ضرورى هو أنه فى العديد من المواقف يقبل المُعرِب جملاً شاذة و غريبة.

الميزة الرئيسية فى هذه الطريقة هى سهولة تنفيذها و خصوصاً فى استخلاص المعلومات من الرسائل بسرعة. و فى الحقيقة، مما لا شك فيه أن أى نظام ناجح لمعالجة اللغات الطبيعية يكون به نظاماً لإهمال و استبعاد الكلمات الغير الضرورية (noise-disposal system).

٥-٤ علم الصرف و القاموس

Morphology and Dictionary

نظم معالجة اللغة الطبيعية (NLP) تستخدم قاموساً لتخزين الكلمات التي يعالجها كمكونات اللغة الطبيعية. و بالطبع لا يمكن تخزين كافة الكلمات بأشكالها المختلفة. و لابد من استخدام علم الصرف الذي يمكننا من ذلك.

Morphology

٥-٤-١ علم الصرف

بالطبع لا يمكننا تخزين جميع كلمات لغة طبيعية أو حتى جزء من لغة طبيعية في قاموس الكلمات فهذا يؤدي إلى نظام غير كفاء. فهناك كلمات بمثابة مصادر للكلمات الأخرى يمكن تخزينها ثم اشتقاق باقي الكلمات من ذلك المصدر فيما يُعرف بعملية اشتقاق الجذور أو الصرف (Morphological Process).

■ مثال على اللغة الإنجليزية

الكلمة perform (يؤدي أو ينجز) التي يمكن أن تأخذ أشكالاً عدة منها على سبيل المثال (performs, performed, performance, performing, etc.) لذلك يمكن اعتبار الكلمة perform جذراً أو مصدراً و نشق منها باقي الكلمات.

■ مثال على اللغة العربية

كلمة زرع التي يمكن أن تأخذ الكثير من الأشكال منها على سبيل المثال (زارع، زراعة، مزارع، مزارعون، وغيرها). لذلك كلمة زرع هي جذر أو مصدر للكلمات الأخرى من خلال عملية الصرف أو الاشتقاق.

Dictionary

٥-٤-٢ القاموس

يحتوى قاموس نظام NLP على المفردات اللغوية التى يعرفها النظام. الوظيفة الأساسية للقاموس هي مساعدة برنامج مُعرِّب الكلمات فى ترجمة (تحويل) الجملة المُدخلة إلى تمثيل داخلى للمعنى ليتم معالجته أى يساعده فى إعراب الكلمات. أى كلمة فى الجملة المُدخلة يجب أن يكون لها (أو لمصدرها) موقع بالقاموس، آخذين فى الاعتبار عملية التصريف التى يجب أن يقوم بها نظام NLP.

بحسب القاموس يتم تقدير قدرات النظام. مشكلة تشكيل و تركيب القاموس ترتبط تماماً بمشاكل تخزين النص. فإذا كان النص مضغوطاً من أجل حجم تخزين أفضل فإن الوقت اللازم للمعالجة سيزيد بسبب عمليات ضغط (compressing) و فك ضغط (expanding) البيانات.

تشكيل القاموس لكل مُدخل بالقاموس (dictionary entry) يعتمد على المعلومات المُخزّنة فى هذا المُدخل. عنصر البيانات الأساسى و الأكثر وضوحاً هو الجذر أو المصدر (Morpheme) و يُطلق عليه الرأس (Head).

كل مُدخل بالقاموس به المعلومات الملائمة. و يكون خوارزم التصريف أو الاشتقاق (Morphological Algorithm) هو المسئول عن فصل أو عزل الرأس (أى الجذر أو المصدر) الموجود فى مُدخلات القاموس عن باقى الكلمات المدخلة.

٥-٥ تطبيق لقواعد النحو الحر

Application on Context-Free Grammar

يُستخدَم الإعراب طبقاً لسياق النحو الحر في معالجة اللغات الطبيعية كما يُستخدَم مع كل لغات برمجة الحاسب تقريباً. على سبيل المثال، نستطيع إعراب لغات مثل لغة C أو C++ أو Lisp وغيرهم طبقاً لسياق النحو الحر. فهو يجعلنا نطبق تقنيات إعراب مفهومة بشكل جيد طُوِّرت أساساً للغات برمجة الحاسب على اللغات الطبيعية.

٥-٥-١ تطبيق الإعراب طبقاً للنحو الحر

Applying Context-Free Parsing

الإعراب طبقاً للنحو الحر له عدة مميزات. أولاً، من السهل برمجته باستخدام لغات البرمجة المنطقية (مثل لغة PROLOG). ثانياً، يمكنه التصرف مع الجملة (sentence) على مستوى الكلمة (word level) و على مستوى العبارة (phrase level). ثالثاً، يعرف أين يكون في الجملة في أى وقت. فعندما نستخدم نحو بسيط مقيّد، من السهل أن نعرّف قواعد انتاج تصف النحو كاملاً.

العيب الرئيسى للإعراب طبقاً للنحو الحر هو أنه لا يستطيع التصرف مع الطرق الصحيحة المتعددة التي يمكن أن تُشكّلها اللغة الطبيعية، بسبب التقيد بالحجم و السرعة. في هذا الجزء نقدم تطبيقاً باستخدام قواعد النحو الحر لجزء من اللغة الإنجليزية. نستخدم في هذا التطبيق طريقة الإعراب من أسفل لأعلى (Top-down parsing).

في الفصل التالي نقدم تطبيقاً كاملاً لفهم اللغة العربية و تمثيل المعرفة المنطقية باستخدام لغة البرمجة المنطقية Prolog.

٥-٥-٢ مثالين للإعراب طبقاً للنحو الحر

Two Examples in Context-Free Parsing

نقدم هنا مثالاً لقواعد النحو الحر لجزء من اللغة الإنجليزية. جدول ٥-١ يعرض أسماء العناصر التي سوف نستخدمها في تمثيل قواعد النحو الحر مع المختصرات المستخدمة. بينما يعرض شكل ٥-٢ قواعد النحو الحر لجزء اللغة المستخدم في المثال. لاحظ أننا استخدمنا مختصرات الرموز في كتابة القواعد و لم نضع الأجزاء اللانهائية بين علامتين "<...>" في وصف طريقة أخرى لكتابة قواعد النحو الحر.

جدول ٥-١ : الرموز المستخدمة في قواعد النحو الحر

الاختصار	الاسم	الاختصار	الاسم
PP	Proportional phrase شبه جملة	S	Sentence جملة
PPS	Prepositional Phrases أشياء جمل	NP	Noun Phrase جملة اسمية
ADJ	Adjective صفة	VB	Verb Phrase جملة فعلية
ADJS	Adjectives صفات	DET	Determiner أداة تعريف
PREP	Preposition حرف جر	VERB	Verb فعل
		NOUN	Noun اسم

1. S	→ NP VP-PPS
2. NP	→ DET ADJS-NOUN
3. ADJS-NOUN	→ ADJ ADJS-NOUN
4. ADJS-NOUN	→ NOUN
5. VP-PPS	→ VP-PPS PP
6. VP-PPS	→ VP
7. VP	→ VERB NP
8. PP	→ PREP NP
9. DET	→ a, the, an
10. NOUN	→ student, man, woman,...
11. VERB	→ study, lesson, course,...
12. ADJ	→ difficult, easy, big,...
13. PREP	→ to, an, on, from,...

شكل ٥-٢ : مجموعة قواعد النحو الحر المستخدمة لجزء من اللغة الإنجليزية.

لاحظ في شكل ٥-٢ أن القاعدتين ٣ و ٤ عبارة عن عملية معاودة تصل في النهاية إلى أن ADJS-NOUN تعادل عدد من الصفات ADJS (من صفر إلى n) يليها اسماً NOUN. كذلك القاعدتين ٥ و ٧ عبارة عن عملية معاودة تصل في النهاية إلى أن VP-PPS تعادل جملة فعلية VP يليها عدداً من أشباه الجمل PPS (من صفر إلى n).

مثال ١

دعنا الآن نطبق جمل قواعد النحو الحر الموجودة في شكل ٥-٢ في إعراب الجملة

The clever Student study the difficult lesson in the Artificial Intelligence.

عملية الإعراب نراها في شكل ٥-٣. لا حظ أننا نستخدم هنا طريقة الإعراب من القاع لأسفل (Bottom-up Parsing).

The clever Student study the difficult lesson in the Artificial Intelligence

DET ADJ NOUN VERB DET ADJ NOUN PREP DET ADJ NOUN



DET ADJ-NOUN VERB DET ADJ-NOUN PREP DET ADJ-NOUNS



NP

VERB



NP

PREP



NP

NP

VP

PP

NP

VP-PPS



S

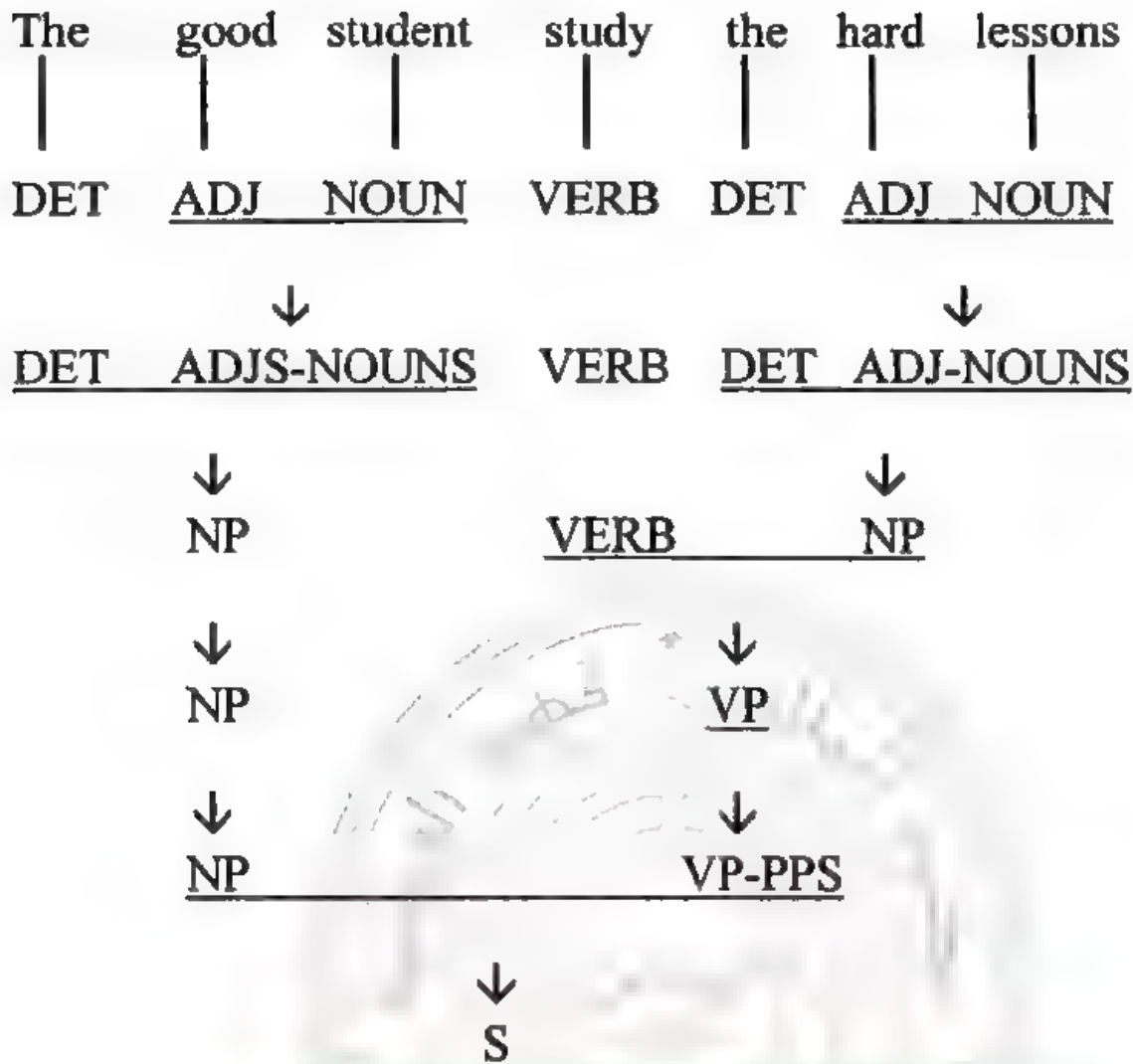
شكل ٥-٣ : عملية إعراب جملة إنجليزية باستخدام النحو الحر

مثال ٢

نستطيع استخدام الشجرة في تمثيل قواعد النحو الحر (Context-Free Grammar) في إعراب الجملة التالية :

The good student study the hard lessons.

الإعراب نراه في شكل ٥-٤، بينما التمثيل باستخدام شجرة نراه في شكل ٥-٥.

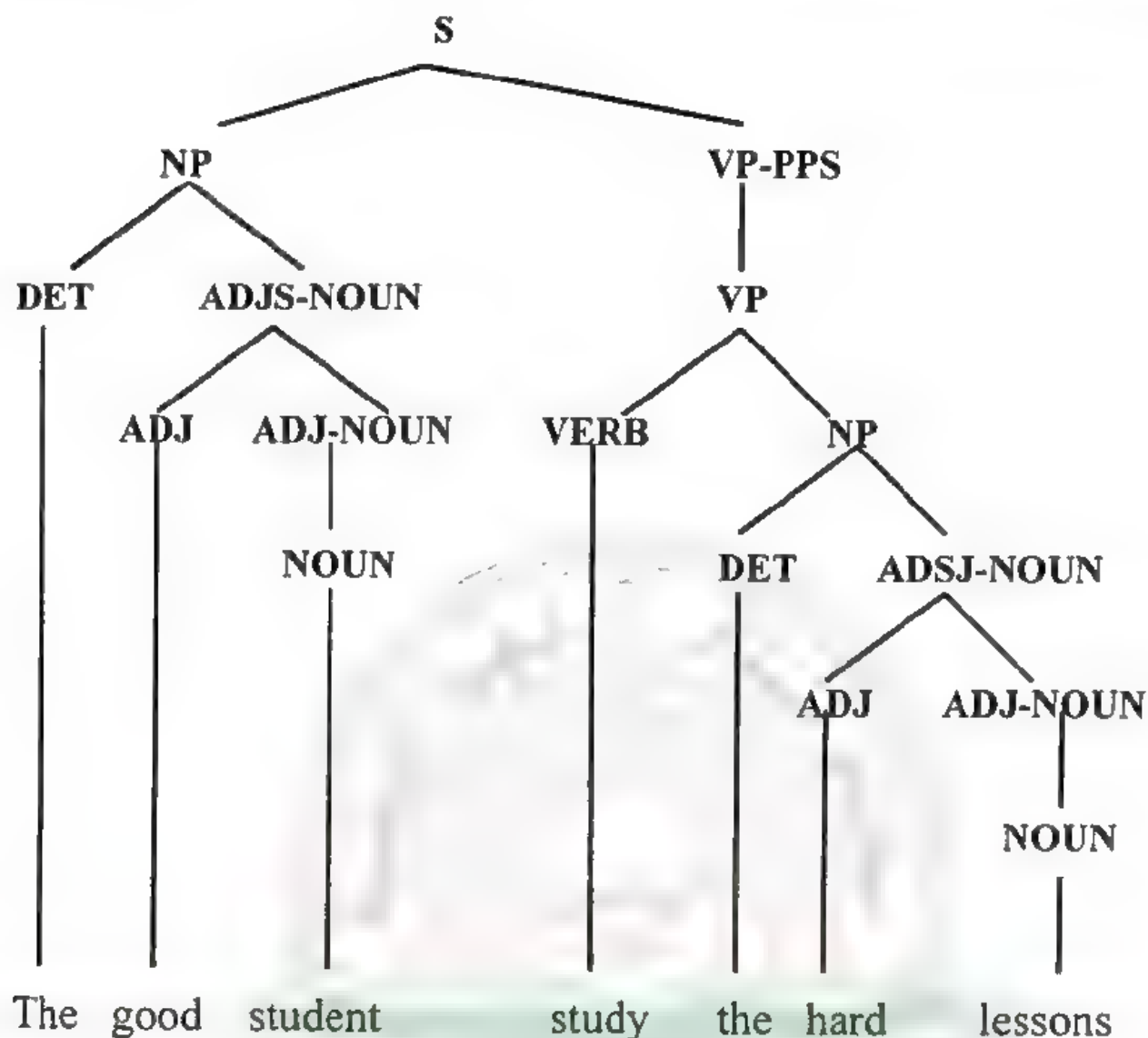


شكل ٥-٤ : إعراب جملة انجليزية.

Questions

٥-٦ أسئلة

١. ما الهدف من معالجة اللغات الطبيعية؟ و ما هي طرق معالجتها؟
٢. ما المقصود بـ Production Rules و وضع استخدامها في عملية الإعراب.
٣. وضع بإيجاز أنواع النحو المختلفة.



شكل ٥-٥ : تمثيل شجرة الإعراب باستخدام الشجرة الثنائية.

٤. عرف مايلي : Parsing, Parser, NLP.

٥. وضح أنواع الإعراب التالية مع عمل مقارنة فيما بينها :

Top-Down Parsing, Bottom-Up Parsing,
and Noise-Disposal Parsing

٦. ما المقصود بكلٍ مما يلي مع بيانات أهميتها في نظم NLP :

Morphological Process, Morpheme, Dictionary, and Head in Dictionary.

٧. باستخدام النحو الموجود في شكل ٥-٣ اعرّب الجمل التالية :

The new player move the heavy box to the table.

A good boy follow the helpful advise from the old man.





الفصل السادس

فهم اللغة العربية و استخدام المنطق الرياضي

Arabic Language Understanding and Using Mathematical Logic

بدأت معالجة اللغات الطبيعية منذ ما يقرب من ثلاثين عام لما لها من أهمية و تأثير فعال. بعد ذلك بقليل بدأت معالجة اللغة العربية كواحدة من أهم و أصعب اللغات الطبيعية تشق طريقها إلى الوجود. و نظراً لأهمية هذا الموضوع نركز هذا الفصل على دراسة كُنه نظم معالجة اللغات الطبيعية (Natural Language Systems) بصفة عامة و نظم معالجة اللغة العربية بصفة خاصة. ثم نتعرض لتطبيق استعمال باللغة العربية الفصحى ((Arabic Question Answering System (AQAS)) كمثال لتلك النظم.

Preface

٦-١ تمهيد

لقد نشأ العديد من تقنيات الذكاء الاصطناعي أساساً كجزء من بحوث معالجة اللغات الطبيعية. و كان الهدف المفترض هو بناء برنامج فاهم اللغة (Language Understander) بحيث يضاعف قدرة البشر العادية على تفسير اللغة الطبيعية و يستخدم الفهم الناتج للأسئلة و يترجم أو يتخذ إجراءً معيناً. عبر السنوات الماضية

طُوِّرت عدة برامج حققت خطوات هامة مثل فهم الحوار و تفسير القرارات و بناء واجهات اتصال باللغة الطبيعية.

تختلف نظم المعرفة عن النظم التقليدية فى أنها تمثل قاعدة معرفة فى شكل عالى المستوى بدلاً من ترميز المعرفة فى جمل منخفضة المستوى. فهى تُخزّن قاعدة معرفة مُكوّنة من حقائق و قواعد. و يوجد نوعان من نظم المعرفة :

- **نظام خبير (Expert System)** مثل نظام حل المشكلة (Problem-Solving System) الذى يصل إلى مستوى الخبير أو على الأقل إلى مستويات عليا من الأداء. (الفصل الثامن يعرض نظام خبرة عام (Nasser96) للتصنيف باستخدام الحالات مُبرمج بلغة C++).

- **نظام معالجة اللغة الطبيعية (NLP System)** الذى يُحدث البشر بجمل لغوية و يتقبّل منهم المعرفة الجديدة و يرد على أسئلتهم بمرونة و يسر. (سوف نعرض فى هذا الفصل نظام فهم اللغة العربية (AQAS) مبنى بلغة PROLOG بغرض الاستعلام من قاعدة بيانات للأمراض الناشئة عن الإشعاع الذرى)

٦-٢ نظام معالجة اللغات الطبيعية

Natural Language Processing System

نظام معالجة اللغة الطبيعية ((Natural Language Processing (NLP) هو نظام مبنى على المعرفة (Knowledge_based System) مُصمّم لفهم و معالجة اللغة الطبيعية. و نستطيع أن نطلق عليه نظام معرفة باستخدام لغة طبيعية (Knowledge_based Natural System). يجب أن يكون هذا النظام قادراً على قبول مُدخلات فى نص لغوى فى مجال تطبيق معين و تخزين المعرفة

المرتبطة بمجال التطبيق و كذلك استنتاج المعرفة و استخلاصها للإجابة على الأسئلة المناسبة و توليد الإجابة للمستخدم.

١-٢-٦ مهام نظم معالجة اللغة الطبيعية

Tasks of NLP Systems

نظم المعرفة الكاملة يجب أن تكون قادرة على أداء المهام التالية :

- تحل أو تساعد في حل المشاكل الهامة التي يمكن أن تتطلب خدمات خبير بشري.
- تدمج المعرفة الجديدة في شكل متزايد في قاعدة المعرفة.
- تساعد المصمم في تنظيم و إنتقال المعرفة.
- تعرض المعرفة في شكل يسهل على المستخدم قراءته و فهمه.
- توفر تفسير لما تصل إليه/و للنصيحة التي تقدمها.
- تبرر حكمها عن طبيعة المهمة التي تؤديها أو طرق تنفيذها بكفاءة.
- تدعم واجهة اتصال طبيعية و مقروءة.

يمكن فقد بعضاً من هذه الصفات في نظام معين و لكن هذه الصفات تخدم كمعايير لقياس عمق نظام المعرفة. و تخدم أيضاً بشكل عام في إيجاز العديد من أهداف البحث في هذا المجال.

٢-٢-٦ مكونات نظم معالجة اللغة الطبيعية

Components of NLP Systems

لدعم المهام التي ذكرناها في الجزء السابق، فإن نظم المعرفة تُبنى من المكونات التالية :

- قاعدة معرفة (Knowledge Base) فى شكل قواعد و حقائق.
- محرك استنتاج (Inference Engine) ليستدل على الحلول باستخدام الحقائق والقواعد الموجودة فى قاعدة البيانات.
- مُولّد (Generator) لتفسير الاستدلال على الحلول.
- طرق استنباط (Acquisition Techniques) لاستخلاص معرفة جديدة و ترميزها فى قاعدة البيانات.
- برنامج للحوار (Dialogue Handler) ربما يتراوح فى الصعوبة من قائمة إلى معالج لغة طبيعية كامل.

أقصى ما نريده هو أن نكون قادرين على الاتصال مع نظام معرفة كما نفعل مع شخصٍ ما. و نرغب فى وضع معرفة جديدة بلغةٍ ما و أن نسأل أسئلة و نحصل على أجوبة باللغة الطبيعية.

٦-٢-٣ مراحل إنشاء نظم معالجة اللغة الطبيعية

Phases of NLP Systems

على الرغم من أن الناس يستطيعوا التعامل مع هذا التنوع، فإن أغلب نظم المعرفة و أغلب نظم معالجة اللغة الطبيعية قد طُوِّرت للاستخدام فى موضوعات مُحدَّدة. حتى عند اختيار موضوع واحد فإن الكثير من تنوع معالجة الجمل سوف يتبع الخطوات التالية :

- تقسيم الجمل إلى قائمة من الكلمات.
- البحث عن كل كلمة فى القاموس، لاكتشاف على سبيل المثال ما إذا كانت الكلمة فعلاً أو اسماً و أى تصريف لغوى بها.

- استخدام نحو (grammar) لغة طبيعية للعثور على إعراب واحد أو أكثر للجملة. الجملة التي يكون لها أكثر من إعراب واحد ينشأ عنها غموض و التباس.
- تحويل الإعراب إلى معرفة داخلية لنظام المعرفة. (في أغلب النظم كون المعرفة منطقية).
- معالجة الشكل الداخلي لجملة. ربما باستخدام أنواع استنتاج منطقي. فإذا كانت الجملة سؤالاً يكون المطلوب إنتاج جواب مناسب لها.
- إذا تم إنتاج إجابة يكون في التمثيل الداخلي بلغة النظام.

الخطوات ٤ و ٣ و ٢ و ١ (أي بطريقة عكسية) تُستخدَم لترجمة الشكل الداخلي إلى مخرج باللغة الطبيعية. هذه الخطوات أقرها مركز أبحاث IBM عام ١٩٨٧ م. فيما يلي نقدٌ تعليقاً على تنفيذ تلك الخطوات :

- الخطوة ١ تكون سهلة التنفيذ في أغلب اللغات الطبيعية، نظراً لأنه علينا أن نبحث عن حروف الفراغ و نقطع الجملة إلى كلمات بناءً على وجود الفراغات.
- الخطوة ٢ ربما يوجد عدة طرق للعثور على الكلمات في القاموس.
- الخطوة ٣ ربما تنتج شجرات إعراب مختلفة، ليست جميعها لها معاني مُتركة بالنسبة للغة الطبيعية.
- الخطوات ٢ و ٣ و ٤ من أجل وضوح المفاهيم ربما تكون مستقلة بدلاً من كونها مختلطة. لكن ذلك ربما يستهلك وقت كبير من الحاسب لأن كل منها ممكن أن ينتج العديد من البدائل التي يتم رفضها من قبل الخطوة التالية، لذلك يتم إجراء الخطوات بشكل متداخل و متعاون.

٦-٣ نظام استعلام باللغة العربية الفصحى

Arabic Question Answering System(AQAS)

إن معالجة اللغة العربية بالحاسب لم تعد رفاهية أو أمراً ثانوياً، بل هي أمر في غاية الأهمية. نقدم فيما تبقى من هذا الفصل وصف دقيق لأحد نظم فهم اللغة العربية و هو نظام AQAS (ناصر، ١٩٩١م و ١٩٩٣).

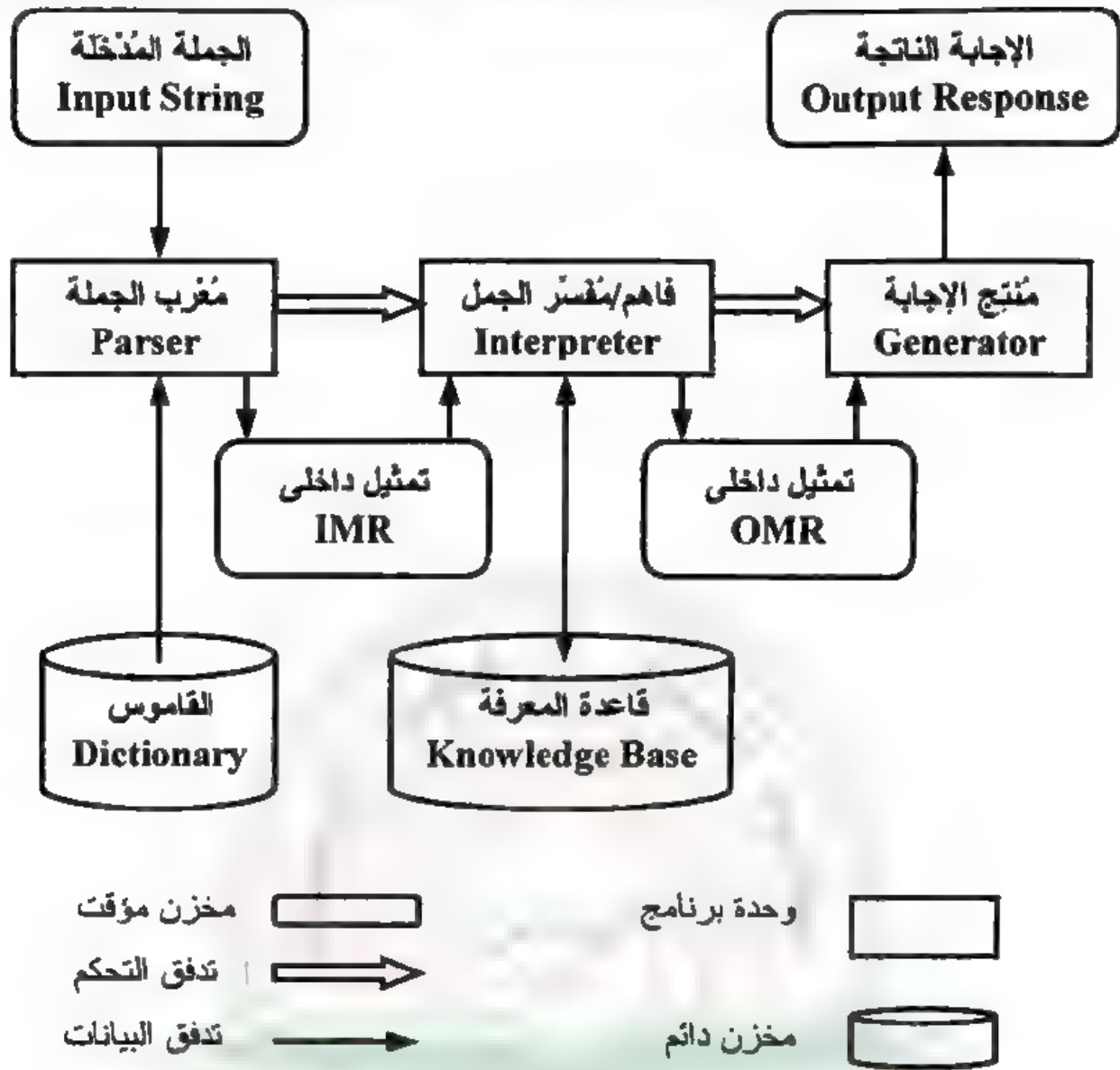
نظام AQAS هو نظام استعلام باللغة العربية (Arabic Question Answering System (AQAS)) و هو نظام مبنى على المعرفة (Knowledge-based System) يقبل جملة مُدخلة باللغة العربية. فإذا كانت الجملة المُدخلة عبارة عن جملة خبرية نَعْلَمُ منها النظام و إن كانت سؤالاً أنتج له النظام الإجابة المناسبة.

٦-٣-١ بناء نظام AQAS Structure of AQAS System

يتكون نظام AQAS كما نرى في شكل ٦-١ من الأجزاء التالية :

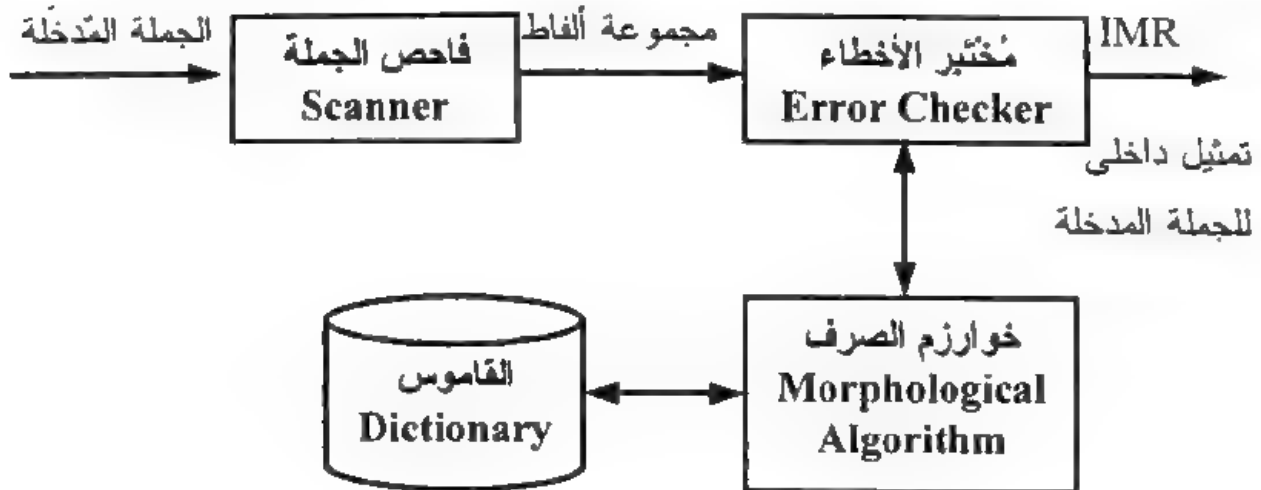
▪ معرب الجمل (Parser)

يتقبل برنامج المُعَرِّب الجملة المُدخلة كنص حرفى (سلسلة حرفية) و يقوم بتقطيعه إلى مجموعة من الألفاظ (tokens) و يقوم بهذه العملية مُتَفَحِّص/ماسح الجملة (scanner). بعد ذلك يقوم مختبر الأخطاء (error checker) باختبار كون كل لفظة جزء من نموذج صحيح مُحدَّد من قِبَل نحو اللغة الطبيعية المُستخدمة.



شكل ٦-١ : رسم تخطيطي لنظام AQAS.

الهدف من هذه المرحلة هو تحويل كل جملة مُدخلة إلى تمثيل داخلي لمعنى المُدخّل (Internal Input Meaning Representation (IMR)) مثل بناء شجرة الإعراب لكي يسمح بالمعالجة. و يتبع البرنامج تشكيلات محددة للجمل المسموح بإدخالها. يقوم المعرب باستدعاء خوارزم التصريف اللغوي لفصل المشتقات حسب تركيب القاموس. شكل ٦-٢ يعرض رسم تخطيطي لمُفحّص الجملة.



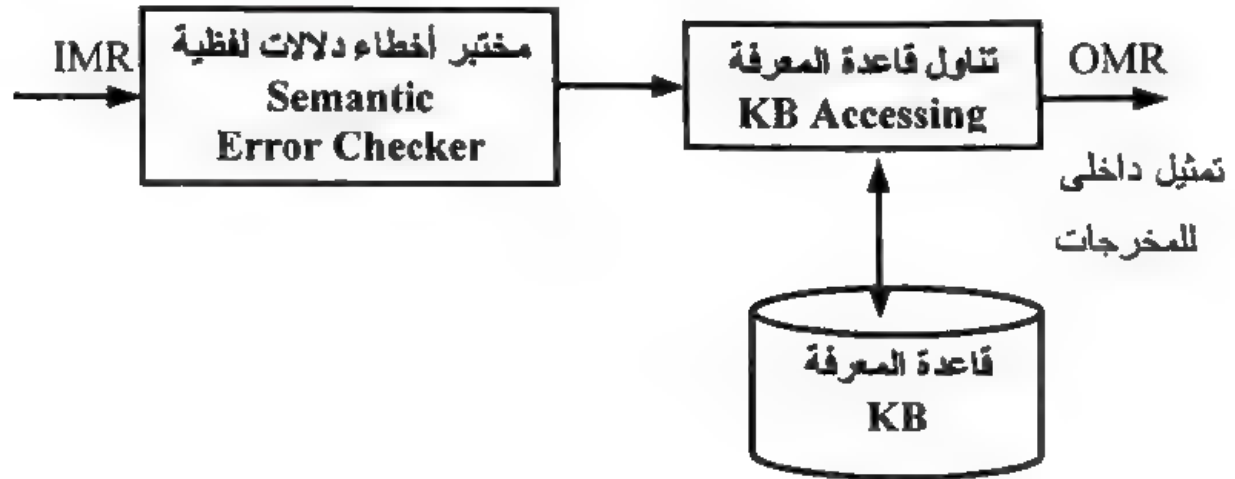
شكل ٦-٢ : رسم تخطيطي لبرنامج المُعرِّب Parser.

■ فاهم الجمل (Interpreter)

هذا البرنامج مسئول عن أى تناول لقاعدة المعرفة (Knowledge Base(KB)) متضمنة أى إضافات أو تعديلات. فوظيفته، كما نرى فى شكل ٦-٣، تعتمد على نوع الجملة المدخلة. فإذا كانت الجملة المدخلة سؤالاً، يقوم البرنامج باستخراج الإجابة المناسبة من قاعدة المعرفة (إذا أمكن ذلك). لكن إذا كانت الجملة المدخلة عبارة عن جملة خبرية (جملة مفيدة) و كانت محتوياتها متوافقة مع محتويات قاعدة المعرفة، قام البرنامج بإضافة المعرفة الجديدة إلى قاعدة المعرفة. يقوم البرنامج ببناء تمثيل داخلي لمعنى المخرجات (Internal Output Meaning Representation (OMR)).

■ منتج الإجابة (Generator)

يشكل برنامج منتج الإجابة شكل المخرجات للمستخدم. فوظيفته، كما نرى فى شكل ٦-٤، تعتمد على مخرج برنامج فاهم الجمل و هو OMR. فهو يتناول OMR و يُظهر الإجابة للمستخدم.



شكل ٦-٣ : رسم تخطيطي لبرنامج فاهم الجمل Interpreter.



شكل ٦-٤ : رسم تخطيطي لبرنامج منتج الإجابة Generator.

■ قاعدة المعرفة (Knowledge Base (KB))

قاعدة المعرفة تمثل ذاكرة نظام AQAS (ماذا يعرف النظام - حقل المعرفة). تحتوي قاعدة المعرفة على معلومات عن أصول الفيزياء الإشعاعية و هي عالم معرفة النظام. تتشكل قاعدة البيانات باستخدام نظام إطار المعرفة (Frame-based System).

■ القاموس (Dictionary)

يحتوي القاموس على المفردات اللغوية للاستعلامات اللغوية العربية (مثل الكلمات الأساسية، الكلمات المطلوب إدراكها و الكلمات المكن إهمالها

خلال عملية الإعراب) و يحتفظ القاموس بخصائص كل كلمة لتستخدم في المعالجة. يتم تصنيف مُدخلات القاموس طبقاً للنحو العربى. فهو يتكون من صنفين : صنف عام (الفعل و الاسم و غيره) و صنف مغلق (حروف الجر و حروف العطف و الضمائر و غيرها).

■ تمثيل المعنى الداخلى (Internal Meaning Representation)

التركيب الداخلى للجمل المدخلة (IMR) و الإجابة الناتجة (OMR) هي الشكل العملى الناتج من المُعرب و فاهم الجملة على الترتيب. حيث يستخدم المُعرب الجملة المدخلة لينتج IMR. ثم يستخدم فاهم الجمل IMR فى إنتاج OMR، بينما يستخدم منتج الإجابة OMR لينتج الإجابة من النظام. هذه التراكيب الداخلية غير مرئية للمستخدم و منفصلة عن القاموس و قاعدة المعرفة.

٦-٣-٢ أنماط و وظائف نظام AQAS

Modes and Functions of AQAS System

للحصول على نتيجة لجملة مدخلة يمر نظام AQAS خلال الحالات التالية :

■ حالة السؤال و الإجابة (Question/Answering Mode)

يُقبل السؤال و يُحوّل إلى IMR ثم يتم تناول KB لحساب الإجابة و بناء OMR الذى يحول منتج الإجابة ليخرج لنا النتيجة. يسمح نظام AQAS بأنواع الأسئلة التالية :

١. أسئلة ترجيحية (verification questions) حيث يسأل المستخدم سؤالاً إجابته بالنفى أو الإيجاب (نعم/لا) للصحيح أو الخطأ على الترتيب.

٢. أسئلة كميّة (quantification questions) حيث يسأل المستخدم عن قيمة عددية (أى كميّة) باستخدام كم عدد أو كم الكمية (how many or how much).

٣. أسئلة تحديد قيم خصائص (feature specification questions) حيث يطلب المستخدم استخراج قيمة خاصية معينة و هذا النوع هو الأكثر استخداماً.

▪ حالة نعلم (Learning Mode) تهتم بإضافة معرفة جديدة إلى المعرفة الموجودة بشرط توافقها مع KB.

٦-٣-٣ مجال المعرفة فى نظام AQAS

Knowledge Domain of AQAS System

مجال المعرفة الذى يتخصص به نظام AQAS هو أصول الفيزياء الإشعاعية. و هى تتضمن التأثيرات البيولوجية للإشعاعات و أصول الذرة و النواة و الإشعاع و مصادر الإشعاع الطبيعية و الصناعية و التعامل معها. يهتم النظام بصفة خاصة بكافة الأمراض التى تنشأ عن التعرض للإشعاع الذرى و التأثيرات المبكرة و المزمنة. كذلك يركز على استخدامات الإشعاع فى المجالات الزراعية و الصناعية و الطبية.

٦-٤ تصميم نظام AQAS Design of AQAS System

لتحقيق الأهداف التى ذكرناها فيما سبق، نصف لغة الاستعلام فى نظام AQAS و نصف نموذج النحو المستخدم فى الاستعلام. حيث ينجز AQAS مهمته من خلال وحداته المختلفة : المُعَرِّب و فاهم الجملة و منتج الإجابة. نجاح تلك الوحدات يعتمد على تركيب القاموس و عملية التصريف اللغوى و قاعدة المعرفة.

٦-٤-١ لغة الاستعلام فى نظام AQAS (المدخلات النصية)

Query Language of AQAS System (Input String)

هذا الجزء يهتم بتحديد مرونة و صعوبة اللغة المستخدمة فى هذا النظام. حيث يمكننا تصنيف الاستعلامات فى نظام AQAS كما يلى :

- **جملة أمر (Imperative Sentence) :** هذا النوع من الجمل يبدأ بفعل أمر لاستخراج المعلومات. و يمكن أن يتبع فعل الأمر جملة فعلية (verb phrase) أو جملة اسمية (noun phrase). أمثلة على الأفعال : "أعطني"، "وضح"، "بين"، "أذكر"، "استخرج". كمثال على جملة الأمر :

أعطني قائمة بالنظائر المشعة للهيدروجين.

- **جملة استفهامية (Interrogative Sentence) :** هذا النوع من الجمل يبدأ بأداة استفهام و تأخذ أحد شكلين للاستعلام :

- **استعلام للتقرير :** تبدأ الجملة بأداة الاستفهام " هل " أو " أ " و يتبعها جملة عربية مفيدة كاملة. هنا يكون المطلوب من النظام تقرير صحة الجملة أو خطأها و لذلك تكون الإجابة بالإيجاب : " نعم " أو بالرفض " لا ". كمثال على استعلام التقرير :

هل العدد الذرى للهيدروجين يساوى ١؟

- **استعلام للاستخراج :** هذا النوع من الجمل يبدأ بأى أداة استفهام غير " هل " و " أ ". و يمكن أن يتبع أداة الاستفهام جملة فعلية (verb phrase) أو جملة اسمية (noun phrase). الغرض من هذا الاستعلام هو استخراج

معلومات من قاعدة المعرفة. أمثلة على أدوات الاستفهام : " ما "، " ماذا "، " أين "، " كيف ". كمثال على جملة استعمال للاستخراج :

ما هي المصادر الطبيعية للإشعاع؟

- **جملة خبرية (Declarative Statement) :** تكون هذه الجملة عبارة عن جملة عربية مفيدة كاملة. هنا يكون المطلوب من النظام إضافة معلومة جديدة من المستخدم إذا كانت متوافقة مع قاعدة المعرفة. هذه الجملة مشابهة لجملة الاستعلام للنقير و لكن بدون أداة الاستفهام. كمثال على الجملة الخبرية :

العدد الذرى للكربون يساوى ٨

٦-٤-٢ النحو فى نظام AQAS

Grammar in AQAS System

بحسب برنامج المُعَرِّب هل يمكن إنتاج سلسلة الألفاظ المُدخلة للنظام بواسطة النحو. حيث يتحكم المُعَرِّب بالاستعلامات التى وصفناها فى الجزء السابق. لذا لا يمكن فصل المُعَرِّب عن النحو فهما وجهان لعملة واحدة.

طبقاً للتصنيف السابق الاستعلامات و لحدود بناء نظام AQAS، فإن القواعد التى تحكم الاستعلامات إلى مجموعات ثلاثة تتوازي مع التصنيفات الثلاثة السابقة. علينا أن نتعرف على مميزات السياق العربى و هى :

- **الجملة** هى العنصر الأساسى لأى لغة، فهى تعبر عن اعتقاد أو فكرة و تتكون من مجموعة كلمات. تحتوى الجمل دائماً على عنصر تتحدث عنه و الأشياء التى نصف بها ذلك العنصر.

■ من الممكن لبعض الكلمات العربية أن تكون فعلاً أو اسماً ، فاعلاً أو مفعول به، مثل " مرض " أو " كاتب " مثلاً. هذا التغيير يرجع إلى موضع الكلمة فى الجملة. أيضاً لفهم معنى كلمة معينة نحتاج إلى علامات التشكيل و هى غير متوفرة فى أغلب استخداماتها. لذلك نعتد على موضع الكلمة فى الجملة.

■ الجملة المفيدة الكاملة يجب أن تبدأ إما بفعل يليه فاعل (جملة فعلية) أو باسم (مبتدأ) يليه اسم آخر أو شبه جملة (خبر). كذلك يمكن أن تكون الكلمة الاسم فاعلاً أو مفعولاً به أو مبتدأ أو خبر أو غير ذلك و فى نفس الوقت ممكن أن تكون مفرداً أو جمعاً، مذكراً أو مؤنثاً. أما الفعل فممكن أن يكون مضارعاً أو ماضياً للمعلوم أو للمجهول أو أمراً.

بصفة عامة، نحو اللغة الهدف فى نظام AQAS لا يخوض فى أنواع الأسماء أو الأفعال و لا يهتم بالتشكيل. حيث يبحث النظام عن المعلومات التى تعطيه القدرة على استخراج المعلومات المناسبة. هذه المعلومات يجب أن تخبر النظام عن الشيء الذى نسأل عنه أى المعلوم (Known) و عما نحتاجه من معلومات أى المطلوب (Required) عن ذلك الشيء المعلوم. أى أن النظام يبحث عن تركيب معين يحدده نحو AQAS. المثال التالى يوضح الفكرة :

ما هى المصادر الطبيعية للإشعاع؟

تتكون الجملة من أداة استفهام "ما" و ضمير "هى" و مبتدأ "المصادر" و صفة "الطبيعية" و جار و مجرور هما "الإشعاع". لفهم هذا الاستعلام و الإجابة عليه لا حاجة لمعرفة تلك المعلومات. لكننا نحتاج إلى معرفة أداة الاستفهام و الشيء المعلوم (Known) و هو "الإشعاع" و ما نريد معرفته (Required) و هو

"المصادر الطبيعية". و هذا ما يُسمَّى النحو الوصفي (Descriptive Grammar) حيث يُسمَّى كل جزء حسب استخدامه أو وظيفته.

كل جملة استعلام تتكون من أجزاء لا نهائية (non-terminals)، و سوف تحل محلها كلمات اللغة العربية التي يعرفها نظام AQAS. تتكون جملة الاستعلام أساساً من الجزء المعلوم Known و المطلوب Required بالإضافة إلى بعض الكلمات الغير ضرورية و أدوات الاستفهام و الضمائر كما نرى في قواعد BNF الموجودة في شكل ٥-٦.

- [1] <sentence> ::= [<noise>] {<qarticle> [<pronoun>] [<conjunction>]
[<pverb>][<preposition>][<quantifier>]<query-phrase><term>}
- [2] <query-phrase> ::= <required> [<conjunction>][<pverb>]
[<preposition>] <known>
- [3] <required> ::= <qname> [<noun>] [<adjective>]
- [4] <known> ::= [<discriminator>] <nominal> [<adjective>]
::= [<noun>] <string>
- [5] <nominal> ::= <proper_name> ; <noun>
- [6] <qname> ::= <noun>
- [7] <discriminator> ::= <noun>

شكل ٥-٦ : أمثلة لقواعد BNF في نظام AQAS.

القواعد الموجودة في شكل ٥-٦ تمثل نموذجاً للقواعد التي يستخدمها برنامج المعرب. العناصر الموجودة بين القوسين المربعين عناصر اختيارية (optional) و ما غير ذلك فهي عناصر إجبارية. لنفهم هذه القواعد دعنا نتأمل المثال التالي:

الجملة الاستعلامية هي : ما هي أعراض مرض الإريثيما؟

ما هي أعراض مرض الإريثيما ؟

<term> <proper_name> <noun> <qname> <pronoun> <qarticle>
<known> <required>

Parsing

٦-٤-٣ الإعراب

يستخدم نظام AQAS طريقة الإعراب من القمة لأسفل (Top-Down Parsing) على قواعد BNF. حيث يحاول المعرب أن يبني شجرة إعراب للنص المُدخَل و يستخدم المعاودة. يحاول البرنامج مطابقة الجملة المدخلة مع قاعدة BNF و هو ما يُطلق عليه Firing. تتم عملية الإعراب تتم من خلال الخطوات التالية :

- يبدأ برنامج المعرب بتنفيذ برنامج مُتفحّص الجملة (scanner or lexical tokenizer) الذى يَقْطَع الجملة إلى مصفوفة من الألفاظ (tokens).
- بعد ذلك تبدأ مرحلة اكتشاف الأخطاء من خلال error-checker الذى يحاول مطابقة مصفوفة الكلمات مع قاعدة BNF معينة. فيبدأ بأول قاعدة فإذا فشل فى مطابقتها يحاول مع القاعدة التالية إلى أن ينجح مع أحد القواعد، فإذا فشل فى العثور على قاعدة مناسبة يرفض الجملة المدخلة و يطلب جملة جديدة.
- فى كل محاولة من محاولات المطابقة السابقة يتناول المعرب مدخلات القاموس للعثور على الكلمات الموجودة فى المصفوفة و استخراج المعلومات المصاحبة للكلمة. و يستخدم الكلمة مع المعلومات المستخدمة فى بناء التمثيل الداخلى (IMR) كشجرة إعراب.
- فى الغالب يستدعى برنامج المُعرب خوارزم التصريف اللغوى (morphological algorithm) قبل البحث عن الكلمة فى القاموس. و سوف نتعرّف على القاموس و التصريف اللغوى فى الجزء التالى.
- عندما ينجح المعرب فى المطابقة مع قاعدة ما، يبني شجرة الإعراب التى تحتوى فى أوراقها على الكلمات الموجودة فى المصفوفة التى تتكون منها الجملة المدخلة.

٦-٥ القاموس و عملية الصرف اللغوي

Dictionary and Morphological Process

نجاح عملية الإعراب يعتمد على استخلاص الشكل الأصلي للفظ (عملية التصريف اللغوي) و على العثور عليها في القاموس و استخراج بعض المعلومات المساعدة من القاموس.

Dictionary

٦-٥-١ القاموس

يعتمد بناء القاموس على محاكاة الإنسان في استخلاص المشتق الأصلي للفظ معينة (عملية التصريف اللغوي)، حيث لا حاجة إلى الرجوع بالكلمة إلى الجذر ثلاثي الحرف. لذلك من الممكن أن تجد مدخلات مختلفة لكلمات لها نفس الجذر، مثل الكلمات المصادر أو الرأس (Head) "تطبيق" و "طبق". و عندما ترد الكلمة "التطبيقات" لا نردها إلى "طبق" و لكننا نردها إلى "تطبيق".

يحتوي مُدْخَلُ القاموس على عمود آخر غير الكلمة المصدر (أي الرأس Head) و نطلق عليه معنى الرأس (Head Meaning) ليرشد النظام إلى نوع المطلوب (required) أو المعلوم (known). كذلك معنى الرأس يرشد القاموس إلى الإطار الذي يمكن أن توجد به المعرفة في قاعدة المعرفة.

يوجد بالقاموس صنفين من المدخلات هما :

- مُدْخَلُ مُفْتَوَح (open category) يحدد الوظيفة الأساسية للكلمة مثل اسمي (nominal) أو فعل/إجراء (action). أغلب كلمات اللغة الطبيعية تنتمي إلى هذا النوع. شكل ٦-٦ يقدم بعض الأمثلة من هذا النوع.

noun (اسم)

structure (التركيب) : e_noun (Head, Head_meaning)

examples : e_noun("أثر", "تأثير").

e_noun("نتيجة", "نتائج").

adjective (صفة)

structure (التركيب) : e_adj (Head, Head_meaning)

examples : e_adj ("ارتفاع", "عالي").

e_adj ("ارتفاع", "باسق").

pverb (فعل ماضى أو مضارع)

structure (التركيب) : e_noun (Head, Head_meaning)

examples : e_noun("نتيجة", "نتج").

e_noun("نتيجة", "ينتج").

شكل ٦-٦ : أمثلة لمدخلات القاموس من النوع المفتوح.

- **مُدْخَل مُغْلَق (closed category)** يتضمن عدداً محدداً من بعض العناصر مثل الضمائر و حروف الجر و اسم استفهام و أدوات الوصل. مُدْخَل من هذا النوع يحتوى إما على نفس المعلومات الموجودة فى مُدْخَل من النوع المفتوح أو على قائمة/مصفوفة من الكلمات التابعة لنوع معين من هذه الأنواع. شكل ٦-٧ يعرض أمثلة من هذا النوع.

Conjunction (أداة وصل)

structure (التركيب) : e_cnonj (HeadList)

examples : e_conj(["الذى", "التي",]).

pronoun (ضمير)

structure (التركيب) : e_pronoun (HeadList)

examples : e_pronoun(["هو", "هي",]).

qarticle (أداة استفهام)

structure (التركيب) : e_qarticle (Head, Head_meaning)

examples : e_qarticle ("أين", "مكان");

e_qarticle ("هل", "تقرير");

شكل ٦-٧ : أمثلة لمدخلات القاموس من النوع المغلق.

الأمثلة الموجودة في شكل ٦-٦ و ٧-٦ تستخدم طريقة لغة PROLOG

المستخدمة في نظام AQAS في عرض التمثيل الفعلي الموجود في قاموس النظام لمدخلات القاموس.

٦-٥-٢ التشكل/التصريف اللغوي Morphological Process

تعتمد عملية التصريف اللغوي بشكل أساسي على مُدخلات القاموس و علم الصرف. يقوم خوارزم التصريف في نظام AQAS بالعمليات التالية :

■ إزالة الإضافات (removing additions)

يزيل النظام أى حروف إضافية ملتصقة بالكلمات ليحصل على الكلمة المصدر (الرأس) التى يُحتمل أن يجدها فى أحد مُدخلات القاموس. بعض هذه الحروف الإضافية يوجد فى بداية الكلمات (prefix) مثل : "ال" و "ل" و "قال" و غيرها، و البعض الآخر موجود فى نهاية الكلمة مثل : "ون" و "ين" و "ات" و غيرها.

■ فصل/عزل الكلمات (disconnecting words)

هنا يقوم النظام بفصل الضمائر و حروف الجر الملتصقة بالكلمات مثل "ه" و "هما" و "ك" و "ب" و "و" وغيرها.

العمليات السابقة تتم من خلال خوارزم التصريف. من الممكن أن يستدعى الخوارزم عملية بحث عن كلمة (رأس) فى القاموس أكثر من مرة فى الحالات التالية :

- الكلمة المدخلة كما هى.
- بعد إزالة الحروف الإضافية من بداية الكلمة.
- بعد إزالة الحروف الإضافية من نهاية الكلمة.
- بعد إزالة الحروف الإضافية من بداية و نهاية الكلمة.
- بعد فصل الكلمة من الحروف و الضمائر المتصلة بها.

٦-٦ تمثيل المعرفة Knowledge Representation

استخدم نظام AQAS لتمثيل المعرفة طريقة إطار/هيكل المعرفة (Frame) كتركيبة بيانات تتضمن المعلومات الخبرية و الإجرائية فى العلاقات السابق تعريفها. و كما نعلن أن مجال قاعدة المعرفة لنظام AQAS هو أصول الفيزياء الإشعاعية (Fundamentals of Radiation Physics).

٦-٦-١ النظام المبنى على الإطار Frame Based System

اختار نظام AQAS الإطار لتمثيل قاعدة المعرفة الإشعاعية من خلال تسلسل هرمي يربط بين الإطارات المختلفة. كل إطار له بداية و نهاية و اسم الإطار (name) و نوعه (type) بالإضافة إلى مجموعة أخرى من العناصر كما نرى في شكل ٦-٨. كل عنصر يُطلق عليه فتحة بالإطار (frame slot) الذي يستخدم لتمثيل المعرفة و يمثل أحد خصائص الإطار.

```

frame
  name      :
  type      :
  if-needed :
  if-added  :
  ..
  .. (الفتحات الخاصة بالإطار)
  ..
frame end

```

شكل ٦-٨ : هيكل الإطار.

الفتحة type تحدد الطبيعة العامة للإطار. كذلك توجد ضمن فتحات الإطار فتحة اسمها if-needed (عند الحاجة) تحتوى على الإجراءات التى يمكن تنفيذها لحساب قيم الفتحات عند الضرورة. و هناك أيضاً الفتحة if-added (عند الإضافة) يرتبط بها إجراء يتم تنفيذه عند إضافة معلومات جديدة إلى KB.

Slot Structure

٦-٦-٢ هيكل الفتحة/الفراغ

تستخدم قاعدة المعرفة بعض تقنيات التمثيل داخل الإطارات : الاسم (name) و القيمة (value) و التعليق (comment) و الإجراء (procedure) كما نرى فى شكل ٦-٩.

slot

name : (integer)
value : (real) or (string)
comments : (string)
procedure : condition (symbol)
 procedure (symbol)

شكل ٦-٩ : هيكل فتحة الإطار.

التعليق هو مجرد نص حرفى يمكن إضافته إلى الفتحة. يمكن إدخال قيم أولية (defaults) فى فتحات الإطار.

Frame Structure

٦-٦-٣ هيكل الإطار

يحتوى نظام AQAS المبنى على الإطار على فئتين من الإطارات :

- إطارات وصفية : تضم أنواع الوصف الفيزيائى للعناصر التى يمكن توقعها عن الطبقات (classes) و الكائنات (objects) و تتضمن أوصافاً مثل الحجم و الشكل و المحتويات و المظهر و المصدر و غير ذلك. أمثلة على الإطارات الوصفية : إطار النظائر المشعة ISOTOPE و إطار تطبيقات الأشعة APPLICATION و غيرهما.
- إطارات إجرائية : تصف المواقف الديناميكية لنشاط ما بتحديد الإجراءات التى تخصمها. فهى تنظم متطلبات الإجراءات. أمثلة على الإطارات

الإجرائية : إطار المرض DISEASE و إطار التأثير الإشعاعي
EFFECT و غيرهما.

٦-٧ مثال تنفيذ استعلام Executing a Query

في جلسة مع نظام AQAS نعرض كيفية معالجة استعلام تم إدخاله إلى النظام مع عرض ما يحدث من خطوات يراها المستخدم أو داخل النظام. و نعرض بعض المقاطع ممن البرنامج و هي مكتوبة بلغة TURBO PROLOG. نوجز تلك الجلسة في الخطوات التالية :

١. إدخال الجملة الاستعلامية : من فضلك ما هي أعراض مرض الإريثيما؟
٢. يقوم برنامج scanner بتقطيع الاستعلام باستخدام مقطع البرنامج :

```
scan (" ") :- !
scan(String1) :-
    front(String1, String2, Word),
    assert(word(Word)), scan(String2)
```

نتيجة هذه الخطوة هي :

المدخل : "من فضلك ما هي أعراض مرض الإريثيما" STRING
المخرج : ["الإريثيما", "مرض", "أعراض", "هي", "ما", "فضلك", "من"] WORDS

٣. يبدأ برنامج الإعراب بالقاعدة الأكثر عمومية في النحو و يحاول أن يوجد مجموعة من القواعد تستطيع توليد الجملة. يختبر المعرب الكلمات الغير ضرورية و يحذفها من المصفوفة WORD بالاستعانة بالقاموس و يستدعي خوارزم التصريف اللغوي و يستخدم مقطع البرنامج التالي الذي يطابق الجملة مع القاعدة رقم ٢ كما يلي :

parse(WORDS, 2, QP) :-

g_qartaticle(WORDS, S1,Q,SQ), g_pronoun(S1,S2,P),
g_conj(S2,S3), g_pverb(S3,S4,PV,SPV),g_prepos(S4,S5),
g_quant(S5,S6,QNT,SQNT),query_phrase(S6,S0,QP),g_term(S0).

كل جزء فى المقطع السابق يأخذ اللفظة المناسبة من مصفوفة الكلمات و يرجع
بالباقى و هكذا حتى لا يتبقى غير علامة الاستفهام و ترجع query_phrase
بالكلمات "أعراض" ، "مرض" ، "الإريثما" لتبدأ الخطوة التالية.

٤. يتم تنفيذ المقطع الخاص التالى الخاص بـ query_phrase و هو :

query_phrase (S6,S0,1) :-

required(S6, SX1),g_conj(SX2),g_pverb(SX2,SX3),
g_prepos(SX3,SX4), known(SX4,S0).

٥. المقطع () required يأخذ الكلمة "أعراض" بينما يأخذ المقطع () known
الكلمتين "مرض" و "الإريثما".

٦. يستخرج النظام من القاموس المعلومات المصاحبة لتلك الكلمات و يعرف
أن عليه استخراج جميع الأعراض.

٧. يقوم فاهم الجمل باستخدام تلك القيم و معرفة الإطار الذى يحتوى على
المعلومات المطلوبة و هو إطار المرض DISEASE نو الاسم الإريثما
بالخاصية عَرَض. يتم ذلك باستخدام المقطع التالى :

process(2,3) :

req(_,SLOT,_FRAME1,_), knw(_,_,NOM,FRAME2,_COND),
eq(FRAME1,FRAME2,FRAME),
frame(FRAME,SLOT,NOM,VAL,COM,COND),
assert(omr(VAL,COM)).

٨. ينتقل التحكم إلى الإطار المطلوب لاستخراج البيانات من خلال المقطع التالي :

```
frame("الإريثيميا", "عرض", "مرض", VAL, COM, COND):-
    name(NCODE, NOM, _), findall(W, appear(NCODE, W, _), VAL),
    findall(W, appear(NCODE, _ , W), COM), !.
```

٩. يستخرج الأمر السابق قيم "التفحيات" و "الحروق" في مصفوفة.
١٠. يقوم برنامج منتج الإجابة بإخراج الإجابة إلى المستخدم.

٦-٨ أسئلة

١. ما المقصود بنظام معالجة اللغة الطبيعية؟
٢. أذكر مهام نظم معالجة اللغة الطبيعية؟
٣. وضّح بإيجاز المكونات الأساسية لنظم معالجة اللغة الطبيعية.
٤. بين المراحل التي يمر بها إنشاء نظم معالجة اللغة الطبيعية.
٥. ما هي وظيفة و مكونات كلاً من :

Parser, Interpreter, Generator, KB, Dictionary,
Morphological Algorithm.

في نظم معالجة اللغة الطبيعية.



الفصل السابع

نظم المنطق الفاهض

Fuzzy Logic Systems

نظم الذكاء الاصطناعي و نظم الخبرة على وجه الخصوص ليست قادرة على إعطاء القرارات أو الأحكام بدقة أو بيقين مطلق. لذلك نشأت تقنيات المنطق الغامض أو الغائم للتعامل مع الغموض و عدم الدقة في البيانات و المعلومات.

Preface

٧-١ تمهيد

عندما يُطالَب منا وصف حالة معينة مثل شكل شجرة مثلاً، فإننا لا نستطيع وصفها بقيم رقمية دقيقة. لكننا نستطيع وصفها بأنها عالية جداً أو منخفضة بعض الشيء و أوراقها على درجة عالية من الاخضرار أو الذبول أى نستطيع ذلك من خلال تصنيفات معينة يمكن تقراوح بين قيمتين عليا (أعلى إمكانية) و قيم دنيا (أقل إمكانية).

عدم الدقة يمكن أن ينشأ من مصادر أو تعاريف أو أخطاء مختلفة أو تقنيات النمذجة. هذه المصادر تنشئ غموضاً في أحكام تطبيقات نظم الخبرة. و يمكن أن يؤدي الحرص في استنباط أو جلب البيانات و التصميم الملائم للتطبيق (إذا أمكن) إلى خفض عدم الدقة في المصادر المذكورة.

فى أغلب الأحيان يجد الخبراء فى مجال معين صعوبة كبيرة فى إعطاء قيم عددية تمثل درجة اعتقاده فى المعرفة. لأسباب كثيرة درجة الاعتقاد يمكن تمثيلها بكفاءة أكبر من خلال التصنيف (أى استخدام أسماء لوصف مجال قيم عددية للمتغيرات).

فى هذه الحالة، ممكن لمطورى التطبيقات أن يستخدموا تقنيات تسمح للخبراء بعمل تصنيفات (مثل ممتاز و جيد و متوسط و ضعيف و ضعيف جداً) لدرجة اليقين بالكيفية التى يعتقدوها فى اشتقاق الحقيقة. من هنا نشأت نظرية المجموعة الغامضة (Fuzzy Set Theory) و المنطق الغامض (Fuzzy Logic) على يد العالم لطفى زاده لتتعامل مع هذه النوعية من المشاكل. تتعامل نظرية المجموعة الغامضة مع القضايا التى لها معنى غامض غير واضح أو غير يقينى.

عندما نصف ظاهرة طبيعية مثل الحرارة، نستطيع التعبير عن درجة الحرارة من خلال تصنيفات بدل من درجة الحرارة نفسها. مثل التصنيفات (بارد جداً و بارد و معتدل و حار و حار جداً). مجموعة التصنيفات تلك تُعتبر مثالاً للتعبير عن المجموعة الغامضة و كل درجة حرارة فعلية تأخذ قيمة مع تصنيف أو أكثر من هذه التصنيفات. و تربط نظرية المجموعة الغامضة رقماً عشرياً محصوراً بين صفر و ١ يعبر عن عضوية عنصر ما فى مجموعة معينة.

كذلك عندما نصف أعراض مرض أصيب به شخص ما فإن الخبير (الطبيب المعالج) يمنح قيمة محصورة بين الصفر و ١ فى مقابل مصطلحات تصف علاقة كل عَرَض مرضى (خاصية للمرض)، أى مدى وجود هذه الأعراض فى المريض محل الدراسة.

مما سبق نجد أن استعمال تقنيات المنطق الغامض يمكن أن يقدم إضافة إلى مجال الذكاء الاصطناعي بإمكانية تعامله مع المعرفة الغير يقينية أو الغامضة.

٢-٧ مفهوم المنطق الغامض Fuzzy Logic Concept

نظم المنطق الغامض تجمع المستوى العالي للمرونة و تمثيل المعرفة في النظم الخبرة و نظم دعم اتخاذ القرار مع القدرة و العمق التحليلي للحسابات الطبيعية. و قد قَدَّم العالم لطفى زاده مفهوم المنطق الغامض ليس كطريقة للتحكم الآلى و لكن كطريقة لمعالجة البيانات عن طريق السماح بعضوية مجموعة أو زمرة جزئية بدلاً من عضوية مجموعة رقمية أو لا عضوية على الإطلاق.

هذا المنهج لنظرية المجموعة الغامضة لم يُطَبَّق على نظم التحكم الآلى حتى السبعينات لضعف إمكانيات الحاسب الآلى. و قد أدرك زاده أن البشر لا يطلبون مُدخلات معلومات دقيقة أو رقمية مع قدرتهم على ضبط التحكم الآلى بشكل ممتاز. فإذا أمكن برمجة متحكمات التغذية العكسية (feedback controllers) لتقبل البيانات المُشوَّهة - المحتوية على بيانات غير ضرورية - (noisy data) و المُدخلات الغير دقيقة (imprecise inputs) فإنها ستكون أكثر تأثيراً و فاعلية و أسهل تنفيذاً.

١-٢-٧ الفكرة المركزية لنظم المنطق الغامض

Central Notion in Fuzzy Logic System

الفكرة الرئيسية في نظم المنطق الغامض أن قيم الحقيقة (truth values) في المنطق الغامض (fuzzy logic) أو قيم العضوية (membership values) في المجموعات الغامضة (fuzzy logic sets) تأخذ قيمة تقع في المدى $[0.0, 1.0]$. بحيث تمثل القيمة صفر الخطأ المطلق (absolute falseness) أو أقل قيمة

ممكنة. بينما تمثل القيمة ١ الصحيح المطلق (absolute truth) أو أعلى قيمة ممكنة.

على سبيل المثال، تأمل الجملة :

"أحمد شخص طويل Ahmad is a tall person".

فلو فرضنا أن طول أحمد حوالي ١٧٠ سم، نستطيع منح الجملة قيمة حقيقة مقدارها 0.80 و نستطيع ترجمتها إلى مصطلحات المجموعة الغامضة كما يلي :

أحمد عضو في مجموعة الأشخاص الطويلة

Ahmad is a member of the set of tall persons

هذه الجملة يمكن كتابتها منطقياً بالشكل التالي :

$$f_{\text{Tall}}(\text{Ahmad}) = 0.80$$

حيث f هي دالة العضوية التي تعمل على المجموعة الغامضة من الأشخاص الطويلة (Tall) و ترجع بقيمة بين 0.0 و 1.0.

٧-٢-٢ مميزات نظام المنطق الغامض

Fuzzy Logic System Features

تجمع نظم المنطق الغامض العديد من المميزات منها :

▪ القوة الذاتية

المنطق الغامض قوى بذاته لأنه لا يتطلب دقة أو مدخلات خالية من البيانات الغير ضرورية أو الغير مطلوبة (noise-free). التحكم في

المُخْرَج عبارة عن دالة تحكم سِلْسِلَة بالرغم من المدى الواسع للمدخلات المتغيرة.

■ قابليته للتعديل بسهولة

نظراً لأن متحكم المنطق الغامض (Fuzzy Logic Controller) يعالج قواعد عرفها المستخدم تحكم نظام التحكم الهدف، فيمكن تعديله بسهولة لتحسين أو تغيير أداء النظام عنوة. حيث يمكن بسهولة دمج حساسات جديدة في النظام بتوليد قواعد تحكم مناسبة.

■ رخيص وسهل البناء والفهم

المنطق الغامض غير محدود بعدد قليل من مُدخلات التغذية العكسية أو مخرج تحكم واحد أو اثنين، فليس ضرورياً أن نقيس أو نحسب عوامل معدل التغيير لكي يتم تنفيذه. بيانات أى حساس توفر بعض التلميحات عن أفعال النظام و ردود أفعال كافية للنظام. يسمح ذلك للحساسات بأن تكون رخيصة و غير دقيقة مما يجعل تكلفة و صعوبة النظام بأكمله منخفضين.

■ نطاق واسع للتحكم الآلى

بسبب التشغيل المبني على القواعد، فإن أى عدد معقول من المدخلات يمكن معالجته (١-٨ أو أكثر) و إنتاج عدد كبير من المخرجات (١-٤ أو أكثر)، على الرغم من أن تعريف القواعد بسرعة يصبح صعباً إذا تم اختيار عدد كبير من المدخلات و المخرجات لتنفيذ وحيد بسبب ضرورة تعريف القواعد التى تعرف علاقتهم ببعضهم البعض. من الأفضل تفسير (تجزئ) نظام التحكم إلى أجزاء و استخدام عدة متحكمات صغيرة بالمنطق الغامض موزعة على النظام، كل له مسئوليات أكثر تقييداً.

■ قابل للاستخدام مع النظم الخطية و غير الخطية

يمكن أن يتحكم المنطق الغامض بالنظم الخطية و غير الخطية (non-linear systems) التى يمكن أن يكون صعباً أو مستحيلاً نمذجتها رياضياً. و هذا يفتح الأبواب لنظم التحكم التى تُعتبر دائماً غير قابلة لتطبيق للتحكم الآلى عليها.

٧-٢-٣ أهمية نظم المنطق الغامض

Necessity of Fuzzy Logic Systems

نظم المنطق الغامض تدمج طريقة IF X AND Y THEN Z البسيطة المبنية على القواعد بدلاً من محاولة نمذجة نظام رياضياً. فنموذج المنطق الغامض يعتمد و يُعَوَّل على خبرة العامل بدلاً من فهمه للنظام فنياً.

مثال ١ : طريقة الاستخدام مع تطبيق دعم اتخاذ القرار

عند وضع قواعد العمل فى نظام دعم اتخاذ القرار نضع شروطاً معينة يلزم تحقيقها فى قاعدة معينة لكى يتم تنفيذ القاعدة. هذه الشروط ممكن أن تكون كما يلى :

If price > 40.00
and demand < 600
and quantity-on-hand > 1200
then profitability – profitability – (QOH*untprice)

و تعنى أنه إذا كان السعر أكبر من ٤٠ و الطلب أقل من ٦٠٠ و الكمية الموجودة لدينا (quantity-on-hand (QOH)) فإن الربحية تقل بمقدار حاصل ضرب الكمية الموجودة لدينا فى سعر الوحدة.

نستطيع استخدام المصطلحات اللغوية (high, low, too-much, verylow) و تعنى على الترتيب (عالي، منخفض، زائد، قليل جداً) فى كتابة قاعدة بالمنطق الغامض كما يلى :

If price is high
and demand is low
and quantity-on-hand is too-much
then profitability is verylow.

مثال ٢ : طريقة الاستخدام مع تطبيق تحكم آلى

بدلاً من التعامل مع التحكم بالحرارة بمصطلحات فنية مثل "SP=500F" أو "T<1000 F" أو "210C <TEMP<220C" فإننا نستخدم المصطلحات التالية :

- "IF (process is too cool) AND (process is getting colder)
THEN (add heat to the process)"

تعنى إذا كانت العملية (مجموعة تشغيل) باردة جداً و آخذة فى البرودة أكثر و أكثر، أضف حرارة إلى العملية.

- "IF (process is too hot) AND (process is heating rapidly)
THEN (cool the process quickly)"

تعنى إذا كانت العملية (مجموعة تشغيل) ساخنة جداً و آخذة فى السخونة بسرعة، قم بتبريد العملية بسرعة.

هذه المصطلحات غير دقيقة و وصفية لما يجب حدوثه. المنطق الغامض قادر على محاكاة التصرفات العملية بكفاءة عالية. فهو يوفر مميزات عديدة تجعله اختيار جيد للعديد من مشاكل التحكم الآلى و هذه هى المميزات :

٧-٢-٤ استخدام نظم المنطق الغامض

Using Fuzzy Logic Systems

يتطلب التحكم الغامض بعض المعاملات الرقمية الدقيقة. لكن القيم الدقيقة لتلك الأعداد لا تكون حرجة عادةً إذا لم يكن مطلوباً أداء سريع الاستجابة جداً في الحالة التي يمكن أن يقدروهم ضبط غير دقيق. نقدم فيما يلي أمثلة لاستخدام المنطق الغامض :

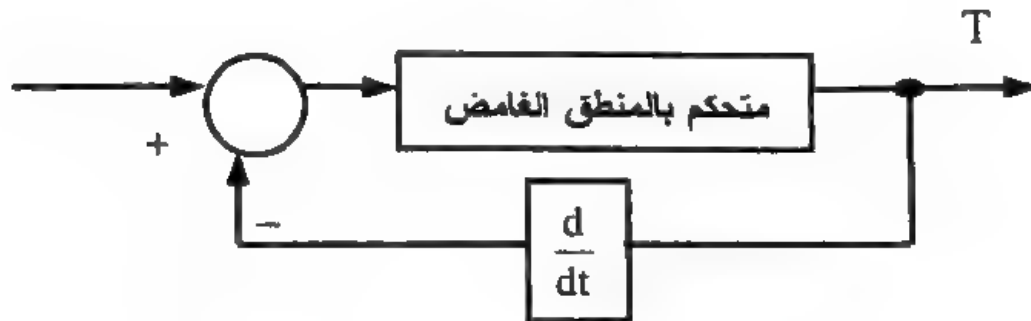
- **النظم الخبيرة (Expert Systems)**

يمكن استخدام المتغيرات اللغوية لتمثيل عوامل تشغيل نظام منطق غامض. و أحد الأمثلة على استخدام المنطق الغامض تقدمه في الفصل الثامن في نظام معرفة خبير في تشخيص الأمراض. في هذا النظام يلزم إدخال أعراض المرض المُصاب به المريض. و عند وصف الأعراض نستخدم المتغيرات اللغوية لوصف بعض الأعراض لتمثل درجة ظهور العرض و علاقته بالمرض و يمكن أن تأخذ أحد القيم (VeryHigh, High, Medium, Low, VeryLow).

- **نظم التحكم الآلى (Automatic Control Systems)**

يمكن لنظام تحكم بالحرارة أن يستخدم حساس حرارة واحد للتغذية العكسية تُطرح بياناته من إشارة الأمر (command signal) لحساب الخطأ (error) مثل التمثيل البسيط الذي نراه في الشكل ٧-١. ثم تتغير مع الوقت تفاضلياً لينتج عنها منحنى ميل الخطأ (error slope) أو معدل تغير الخطأ (rate-of-change-of-error) و يطلق عليه بعدئذ (error-dot). يمكن أن يكون الخطأ وحدات من درجة الحرارة و يتراوح الخطأ بين قيمة صغرى و قيمة كبرى. بعد ذلك يُحسب معدل التغير بالدرجات لكل دقيقة أو

ثانية. نقدم في الجزء ٣-٧ نظاماً للتحكم الآلى باستخدام المنطق الغامض لتحسين التحكم فى عمود تقطير.



شكل ٧-١ : تمثيل بسيط لعملية تحكم آلى.

▪ نظم قواعد البيانات (Data base System)

يمكن استخدام المتغيرات اللغوية فى أوامر SQL عند استخراج البيانات بشروط غامضة من قواعد البيانات. حيث لا يتم تحديد قيمة معينة كحدود قصوى أو دنيا كشروط لاستخراج البيانات. نقدم فى الفصل الحالى (٧-٤) تطبيقاً لاستخدام المنطق الغامض عند استخراج البيانات باستخدام أوامر SQL.

٣-٧ مُتَحَكِّمٌ بِالْمَنْطِقِ الْغَامِضِ لِعَمُودِ تَقْطِيرٍ

Fuzzy Logic Controller for a Distillation Column

التحكم باستخدام المنطق الغامض (Fuzzy Logic Control) هو أسلوب نظام تحكم آلى لحل المشاكل يعين فى تنفيذ النظم التى تتراوح من متحكمات صغيرة جداً وضمنية إلى نظم تحكم ضخمة ذات شبكة حاسبات شخصية و نهايات طرفية لجلب البيانات. يمكن تنفيذ المنطق الغامض باستخدام البرامج أو الأجهزة أو الاثنين معاً. و هى توفر طريقة سهلة إلى استنتاج واضح محدد مبنى على معلومات مُدخلة غامضة أو فيها التباس أو غير دقيقة أو حتى مفقودة.

يتم تطبيق برمجيات المنطق الغامض فى العديد من عمليات التحكم الآلى. هذه النظم المبنية على المعرفة تدمج المعرفة البشرية فى قاعدة المعرفة التى تعمل عليها من خلال قاعدة لقواعد المنطق الغامض و دوال عضوية المنطق الغامض.

عمود التقطير (Distillation Column) هو عملية موجودة فى أغلب الوحدات الصناعية الكيميائية. فى التطبيق الحالى نبين كيفية استخدام المنطق الغامض فى نظم التحكم الآلى (ناصر، ٢٠٠٢).

عمود التقطير يحتوى على عدة دوائر تحكم (control loops) يتم التحكم بها باستخدام متحكمات PID. فى هذا العمل تم تصميم متحكم منطق غامض ليحل محل متحكم PID لضبط درجة الحرارة داخل عمود التقطير. و قد تم تصميم قاعدة تحسوى على قواعد للمنطق الغامض (Fuzzy logic rule base)، يعمل عليها محرك الاستدلال (inference engine) لإنجاز الحل الأفضل.

٧-٣-١ تقديم للتطبيق Introduction To the Application

المنطق الغامض تقنية قوية تسمح للمصممين بدمج الخبرة الهندسية فى إنتاجهم بسرعة. فهى تُستخدَم لإنجاز أداء أفضل فى التحكم فى العمليات. استخدامات المنطق الغامض الأكثر انتشاراً هى تطبيقات التحكم بالمنطق الغامض. و نظام التحكم هو نظام ذو دائرة تحكم مُغلقة يتحكم بآلة أو عملية لإنجاز الاستجابة المطلوبة نتيجة لعدد من مدخلات النظام.

تقوم دائرة التحكم بتحويل المدخلات الرقمية الطبيعية إلى قيم تتناسب و المنطق الغامض أى عضوية المجموعة الغامضة و يُطلق على هذه العملية fuzzification. تتحول مدخلات النظام إلى مدخلات غامضة (fuzzy inputs) لتدخل بدورها إلى محرك الاستدلال (inference engine) الذى يحدد

الإجراء المناسب اتخاذه. تُسْتَخْدَمُ عملية عكسية لما سبق يُطْلَقُ عليها **defuzzification** لتوليد قيم رقمية طبيعية لمخرجات النظام التي تتبع المنطق الغامض.

متحكمات المنطق الغامض متحكمات PID لا خطية تُحَسَّبُ معاملاتها بناءً على إشارة الخطأ و تفاضلها الزمني. في نظم المنطق الغامض توجد المعرفة في أكثر من تمثيل مثل قواعد و مجموعات المنطق الغامض. القواعد تأخذ عناصر البيانات و توجد عضويتهم في المجموعات الغامضة (القاعدة يمكن أن تنشئ مجموعة غامضة إذا كان ضرورياً).

التقطير (Distillation) هو عملية إنتاج بخار (vapour) من سائل بتسخين خليط إلى درجة الغليان ما يؤدي إلى فصل عناصر الخليط عنصر-عنصر بجمع البخار عند درجات حرارة مختلفة دونما أي تحويل للعناصر. التطبيق الذي نعرضه هنا طبقاً على عمود تقطير (packed type distillation column) يُسْتَخْدَمُ للفصل الثنائي (binary separation) لخليط من الكحول و الماء.

الهدف من تصميم متحكم منطق غامض هو حفظ درجة حرارة الخليط عند ٨٠° على قدر الإمكان على الرغم من تدفق الخليط إلى داخل عمود التقطير و تدفق المنتجات إلى خارجه. يتضمن العمود عدة دوائر تحكم للتحكم في الطاقة الداخلة و الخليط الداخل و تدفق المنتجات. و يمكن ضبط دوائر التحكم تلك يدوياً من لوحة تحكم أو آلياً من متحكم PID.

٧-٣-٢ وصف التحكم بالعملية Process Control Description

تم تصميم المتحكم للتحكم في درجة حرارة الخليط (المتغير المُتَحَكَّمُ به) عن طريق التحكم في طاقة التغذية الواردة (المتغير المُتَنَاول). تُضَبِّطُ الطاقة المُدْخَلَةُ حسب

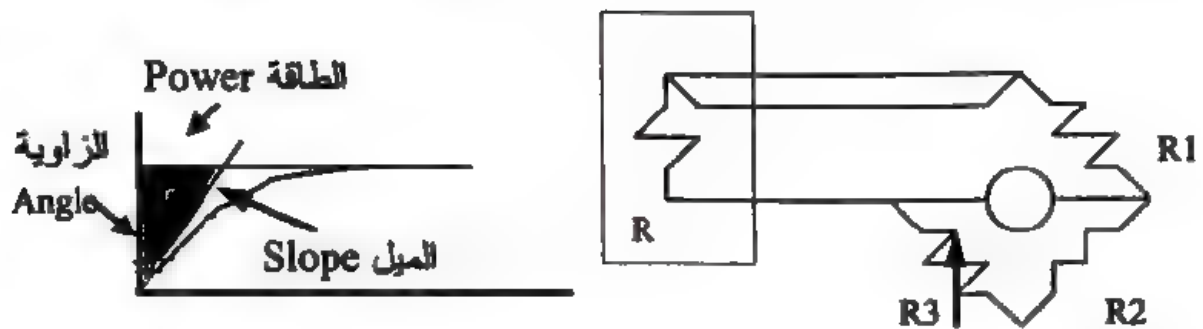
درجة الحرارة داخل العمود و كمية تدفق التغذية الواردة. عنصر التحكم النهائي هو جهاز التحكم بالتسخين (thyristor).

يوجد أربع سخانات (heaters) عند قاع العمود تقوم بتسخين الخليط الوارد إلى العمود. يتم جمع البخار بواسطة مكثف عند قمة العمود حيث ينقسم إلى جزئين: الأول هو المنتج النهائي الخارج (الكحول) و الباقي يرجع إلى العمود لإعادة تبخيره مرة أخرى.

الطاقة القادمة إلى خليط العمود تأتي من ٦ سخانات في العمود، و تُستخدم لغلي الخليط و طاقة كل منها ٦ كيلو وات. كل سخان متصل على التوالي مع جهاز التحكم بالتسخين (thyristor) عند لوحة التحكم التي تتحكم بالطاقة الواردة من السخانات. يتم ذلك عن طريق ضبط زاوية الإحتراق (firing angle) لجهاز التحكم بالتسخين، و بالتالي التحكم في التسخين.

نرى في شكل ٧-٢ قنطرة من أربع أذرع نكتشف الحرارة داخل العمود، كل ذراع عبارة عن مقاومة (platinum thermo couple). عند تغير الحرارة تتغير قيمة المقاومة و كذا قيمة الجهد الكهربائي و بذلك يحس الحساس بالحرارة و يقيسها.

كما نرى في شكل ٧-٣ فإن خصائص جهاز التحكم بالتسخين تشير إلى أن المنطقة المظللة تمثل طاقة جهاز التحكم بالتسخين و هي ميل المنحنى المعتمدة على و المتحكم بها من قبل الدائرة الموجودة في شكل ٧-٢.



شكل ٧-٢: مُكتشف الخطأ شكل ٧-٣: زاوية احتراق المتحكم بالتسخين.

٣-٣-٧ المتحكم بالمنطق الغامض Fuzzy Logic Controller

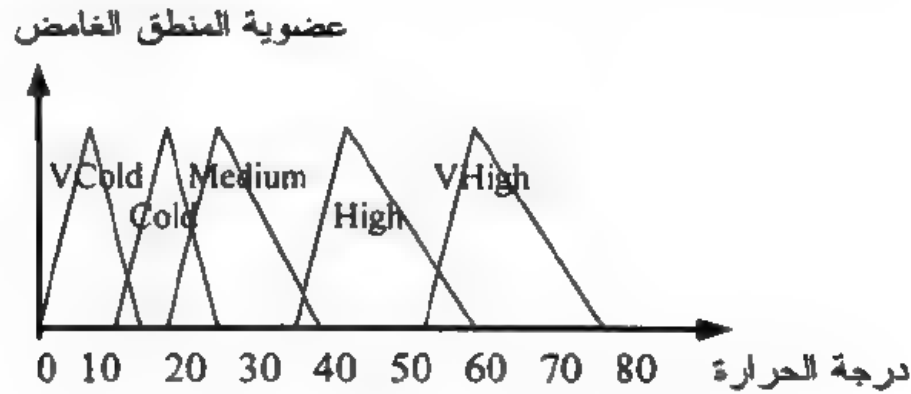
المتحكم بالمنطق الغامض المُقترح نراه في شكل ٧-٤. و كما نرى يوجد مُدخلان : درجة حرارة الخليط داخل عمود التقطير و كمية تدفق الخليط الواردة. أيضاً نرى مُخرج واحد فقط و هو ضبط مقدار الطاقة الواردة إلى العمود.

■ المدخلات و المخرجات (Inputs and Outputs)

مجالات قيم مُدخل و مُخرج المتحكم تأخذ قيم رقمية نراها في جدول ٧-١. يعرض جدول ٧-٢ و شكل ٧-٥ منح المجالات ودوال عضوية المنطق الغامض ذات شكل المُثلث لدرجة الحرارة المُقاسة بواسطة حساس الحرارة (متغير مُدخل رقم ١).

جدول ٧-١ : مدخلات و مخرجات التحكم بالمنطق الغامض

الاسم	النوع	القيمة الصغرى	القيمة العظمى
درجة الحرارة المقاسة	مُدخل	صفر	٨٠ ° مئوية
كمية تدفق الخليط	مُدخل	صفر	٢٠ ميللى أمبير (الإشارة المقاسة)
ضبط مقدار الطاقة	مُخرج	صفر	٦ كيلو وات

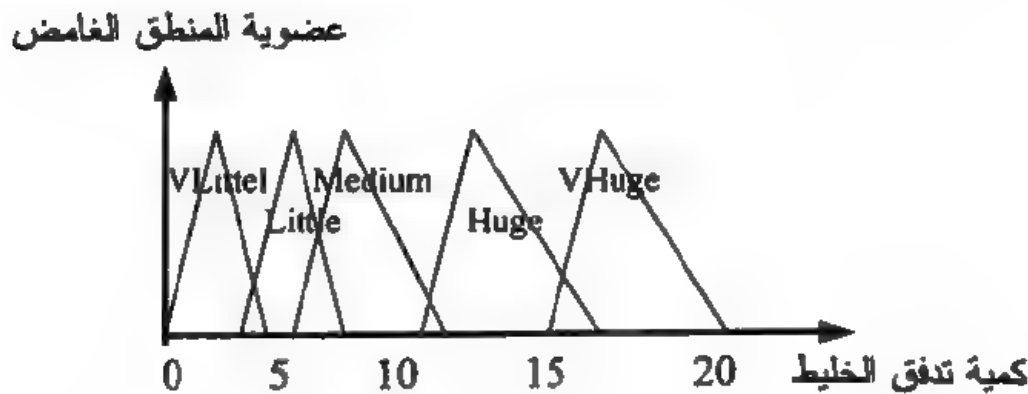


شكل ٥-٧ : دوال عضوية المجموعة الغامضة لدرجة الحرارة.

أيضاً، يعرض جدول ٣-٧ و شكل ٦-٧ منح مجالات و دوال عضوية المنطق الغامض لكمية الخليط الداخلة (متغير مُدخل رقم ٢).

جدول ٣-٧ : مجالات التغير بالمنطق الغامض لكمية الخليط الواردة.

قيمة المتغير الغامض	مجال القيمة الرقمية الداخلة
VLittle - قليل جداً	٥-٠
Little - قليل	٩-٤
Medium - متوسط	١٣-٧
Huge - كبير	١٧-١١
VHuge - كبير جداً	٢٠-١٥



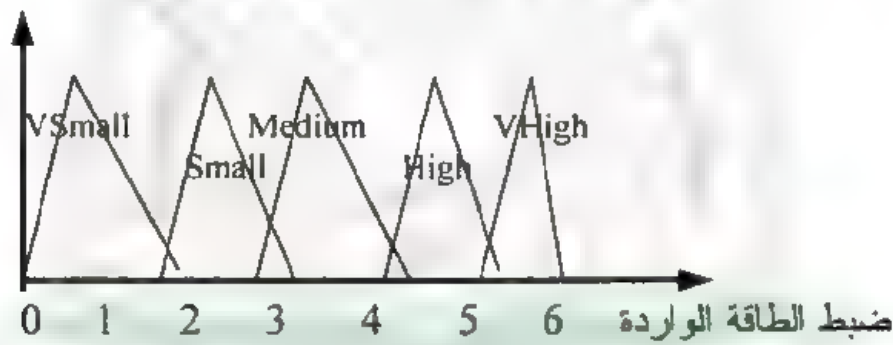
شكل ٦-٧ : دوال عضوية المجموعة الغامضة لكمية تدفق الخليط الوارد.

كذلك يعرض ٧-٤ و شكل ٧-٧ منح المجالات و دوال عضوية المنطق الغامض لضبط درجة الطاقة الواردة (متغير مُخرَج).

جدول ٧-٤ : مجالات التغير بالمنطق الغامض لضبط الطاقة الواردة.

مجال القيمة الرقمية الداخلة	قيمة المتغير الغامض
٢-٠	صغير جداً VSmall
٣,٥-١,٥	صغير Small
٤,٥-٣	متوسط Medium
٥,٥-٤	كبير High
٦-٥	كبير جداً VHigh

عضوية المنطق الغامض



شكل ٧-٧ : دوال عضوية المجموعة الغامضة لدرجة الحرارة.

■ محرك الاستدلال (Inference Engine)

هذا البرنامج يستخدم قواعد IF ... THEN الشرطية الموجودة في قاعدة معرفة النظام للاستدلال على الحلول.

■ قاعدة قواعد المنطق الغامض (Fuzzy Rule Base)

عرّفنا فسمما سبق المدخلات و المخرجات بصيغة المتغيرات الغامضة و لكن ما هي الإجراءات التي يجب اتخاذها وفي أي ظروف. لقد تم تعريف

مجموعة من القواعد التى تصف تشغيل المتحكم. هذه القواعد تأخذ عادة شكل قواعد IF-THEN و يمكن الحصول عليها من الخبراء البشرى فى هذا المجال. جدول ٧-٥ يوضح هذه القواعد. هناك بعض الإرشادات التى يجب مراعاتها عند استعراض المصفوفة الموجودة فى جدول ٧-٥ وهى :

- عندما تكون الحرارة منخفضة يجب ضبط الطاقة الواردة لتكون عالية عن الوضع الذى تكون فيه الحرارة عالية.

- عندما تكون كمية تدفق الخليط قليلة لا نحتاج إلى زيادة ضبط الطاقة الواردة كما هو الحال عندما تكون كمية تدفق الخليط كبيرة.

جدول ٧-٥ : جدول ضبط الطاقة الوارد بناءً على درجة الحرارة و كمية تدفق الخليط.

VCold	Cold	Medium	High	VHigh	درجة الحرارة ← تدفق الخليط ↓
Medium	Small	VSmall			VLittle
High	Medium	VSmall	VSmall		Little
VHigh	High	Small	VSmall		Medium
VHigh	High	Medium	Small		Huge
VHigh	VHigh	High	Medium		VHuge

للحصول على ضبط رقمى للتحكم فى مقدار الطاقة الحرارية الواردة (أى ضبط زاوية الاحتراق)، يتم تنفيذ عدة قواعد IF-THEN من قاعدة قواعد المنطق الغامض فى الحال لأن المدخلات تم تحويلها من أرقام طبيعية إلى قيم منطق غامض (fuzzification). كل قاعدة سوف تنتج قيمة إجراء. بعد ذلك تأتى خطوة تحويل قيمة الإجراء التى تنتمى إلى المنطق الغامض إلى قيم رقمية طبيعية (defuzzification) فنحصل على مخرج المتحكم بالمنطق الغامض. من الجدول ٧-٥ نستطيع استنتاج ٢٥ قاعدة شرطية، نقدم فيما يلى أمثلة منها.

**IF Temperature is Vcold and Flow-Quantity is VLittle
THEN Power is Medium.**

**IF Temperature is Cold and Flow-Quantity is VLittle
THEN Power is VSmall.**

**IF Temperature is VCold and Flow-Quantity is Little
THEN Power is High.**

**IF Temperature is Cold and Flow-Quantity is Little
THEN Power is Medium.**

**IF Temperature is Medium and Flow-Quantity is Little
THEN Power is VSmall.**

**IF Temperature is Vcold and Flow-Quantity is Medium
THEN Power is VHigh.**

**IF Temperature is Cold and Flow-Quantity is Medium
THEN Power is High.**

٧-٤ استخدام المنطق الغامض مع أوامر SQL

Using Fuzzy SQL Commands

سنعرض لاستخدام المنطق الغامض بشكل موجز و سريع لنبين مدى إمكانية الاستفادة من المنطق الغامض في بناء برنامج معالج يقوم بتنفيذ أوامر SQL التي تستخدم المنطق الغامض. سوف يقوم هذا البرنامج بقبول إدخال أمر SQL الذي يحتوى على غموض و يستخدمها في استخراج بيانات من قواعد البيانات بطريقة أكثر كفاءة.

تأمل الجدول ٦-٧ الذى يعرض مثالا لجدول أعضاء هيئة التدريس و هو أحد الجداول المستخدمة فى تمثيل بيانات قاعدة بيانات الجامعة. هذا الجدول يحتوى على الأعمدة/الخصائص : رقم عضو هيئة التدريس و اسمه و درجته العلمية و راتبه و عدد سنوات الخبرة فى الدرجة الوظيفية الحالية و رقم القسم الذى يعمل به. يحتوى الجدول على سجلات بعض أعضاء هيئة التدريس كأمثلة.

جدول ٦-٧ : عضو هيئة التدريس (INSTRUCTOR)

Inst# (رقم المدرس)	Name (الاسم)	Rank (الدرجة)	Salary (الراتب)	Experience (الخبرة)	Dept# (رقم قسم)
676	Khadiga	Prof	15000	5	2
454	Nasser	Asst_Prof	8010	6	2
898	Fahd	Assoc_Prof	10000	2	2
955	Alla	Asst_Prof	7000	5	4

نستطيع استخراج البيانات الموجودة فى الجدول باستخدام أمر استخراج البيانات فى لغة SQL و هو الأمر SELECT. على سبيل المثال، إذا أردنا أن نستخرج الأسماء و الدرجات الوظيفية لأعضاء هيئة التدريس ذوى الرواتب المنخفضة وعدد سنوات الخبرة العالية. قد يقول قائل أن الرواتب المنخفضة هى الرواتب التى تقل عن أو تساوى ٨٠٠٠ و سنوات الخبرة العالية هى ما يزيد عن ٤، فنكتب أمر SQL كما يلى :

```

SELECT  Name, Rank
FROM    INSTRUCTOR
WHERE   Salary <= 8000
AND     Experience > 4 ;
    
```

هذا الأمر سوف يستخرج سجلاً واحداً لعضو هيئة التدريس الذي يقل راتبه عن ٨٠٠٠ و عدد سنوات الخبر أكثر من ٤ هو :

955	Alla	Asst Prof	7000	f	4
-----	------	-----------	------	---	---

لكن ربما يكون هذا الرقم صحيحاً الآن و بعد فترة تزداد أو تُخفّض الرواتب لجميع الموظفين مما يجعل الرقم ٨٠٠٠ ليس معياراً للرواتب المنخفضة. كذلك يمكن أن لا يكون قد تم إدخال جميع السجلات و عند تسجيل باقى السجلات يتغير المعيار أيضاً. كذلك، بالنظر إلى جدول ٦-٧ نجد أنه لو أن راتب عضو هيئة التدريس Nasser هو ٨٠١٠ و هو منخفض أيضاً طبقاً للمعيار المستخدم، و مع ذلك لا نستطيع استخراجه.

لكن إذا استخدمنا مجموع المنطق الغامض (VeryHigh, High, Medium, Low, VeryLow) للتعبير و كان ممكناً أن يتم تغيير مجال كل منهم حسب الرواتب المسجلة بين أقل راتب و أعلى راتب، نستطيع كتابة أمر SQL باستخدام المنطق الغامض كما يلي :

```
SELECT Name, Rank
FROM INSTRUCTOR
WHERE Salary is Low
AND Experience is High ;
```

من المتوقع أن يتم استخراج سجلات أخرى غير الذى حصلنا عليه فى الحالة الأولى، و ربما تكون نتيجة تنفيذ أمر SQL السابق كما يلي :

454	Nasser	Asst Prof	8010	6	2
955	Alla	Asst Prof	7000	5	4

٥-٧ أسئلة

١. ما المقصود بكلٍ من :

Fuzzy Logic, Fuzzy Set Theory, Fuzzification,
Fuzzy Logic Controller, Defuzzification.

٢. ما هو مفهوم المنطق الغامض؟

٣. ما هي الفكرة المركزية لنظم المنطق الغامض؟

٤. حدد أهمية استخدام المنطق الغامض.

٥. أذكر بعض استخدامات المنطق الغامض.

٦. أذكر بعض ميزات نظم المنطق الغامض.

٧. اختار بعض المتغيرات في مجالٍ ما ثم قم بعمل fuzzification لهذه المتغيرات.

٨. اختار تطبيقاً من البيئة التي تحيط بك يحتوي على دوائر تحكم مغلقة و حاول تصميم متحكم بالمنطق الغامض لهذه الدائرة مماثلاً لما درسته في الفصل الحالي.



الفصل الثامن

النظم الخبيرة

Expert Systems

الإنسان الخبير وحده من يستطيع تقديم أداء رفيع المستوى حسب خبرته في مجال تخصصه. ولقد جاءت النظم الخبيرة أو نظم الخبرة لتُجمّع و تستخدم معلومات و خبرة مُركّزة لخبير أو أكثر في مجال معين. يركز أداء النظم الخبيرة في محاكاة أداء الإنسان الخبير في نفس المجال. جاءت أهمية النظم الخبيرة من عدم توفر الخبراء في مجال معين بعدد كاف لاحتياجاتنا. ولقد تم تطوير العديد من النظم الخبيرة في المجالات الدينية و التعليمية و الطبية و الزراعية و الصناعية و الاقتصادية.

Introduction

٨-١ مقدمة

النظام الخبير أو نظام الخبرة (Expert System) هو برنامج يحاكي أداء الخبير البشري في مجال خبرة معين، أو هو برنامج يحل المشاكل التي غالباً ما يستطيع الإنسان الخبير في ذات المجال أن يحلها.

Expert Systems Concept

٨-١-١ مفهوم النظم الخبيرة

النظم الخبيرة هي برامج ذكية تحتوي على الكثير من المعرفة و الخبرة التي يمتلكها خبير أو عدة خبراء في أحد مجالات المعرفة. فهي تستعمل قوانين

التفكير من المنطق والحس العام وغيرها للوصول إلى نتائج مبنية على المعرفة السابقة المخزنة في قاعدة المعرفة.

الميزة الأساسية في النظم الخبيرة هي الفصل بين قاعدة المعرفة (knowledge base) و مُحَرِّك الاستدلال (inference engine). قاعدة المعرفة تضم كافة أشكال المعرفة المتعلقة بتطبيق معين أى معرفة النظام بمجال التطبيق (خبرة النظم الخبيرة). أم محرك الاستدلال فهو البرنامج المسئول عن البحث في قاعدة المعرفة عن الحقائق و القواعد المناسبة و كذا استنتاج حقائق جديدة و استخدام المعرفة القديمة و الجديدة للاستدلال على الحل أى الوصول إليه (reasoning).

٨-١-٢ مميزات النظم الخبيرة Expert Systems Features

تختلف النظم الخبيرة عن البرامج التقليدية في كثير من العناصر، أهمها التفكير. على الرغم من القناعة التامة بأن النظم الخبيرة لن تحل محل الخبرة البشرية، إلا أنها تتميز بالعديد من الصفات منها :

- سهل الاستخدام لأى مستخدم عادى غير الخبير أو المَطَوَّر.
- نافعاً في مجال التطبيق بشكل واضح.
- قادراً على التعلم من الخبراء بطريقة مباشرة وغير مباشرة.
- قادراً على تعليم غير المتخصصين.
- قادراً على تفسير أى حلول يصل إليها مع بيان كيفية الوصول إليها.
- قادراً على الاستجابة للأسئلة البسيطة أو الأسئلة المعقدة في حدود التطبيق.
- وسيلة مفيدة لتوفير مستويات عالية من الخبرة في حالة عدم وجود خبير.
- قادراً على تطوير أداء المتخصصين ذوى الخبرة البسيطة.

مجال النظم الخبيرة واسع و مفيد جداً، لكنه يتضمن بعض المشاكل :

- ذات تكلفة عالية مقارنةً بالتطبيقات التقليدية.
- نطاق تطبيقها محدود في النظم الإدارية و استرجاع المعلومات المتكاملة.
- الأسباب التي تدعونا لبناء نظم خبيرة هي :
- الاحتفاظ بالخبرة و المعرفة من الاندثار أو الانقراض، و خصوصاً في التخصصات الهامة الكثيرة الاستخدام أو النادرة.
- حل المشاكل، مما يحفظ الوقت و المال و الجهد.
- زيادة الخبراء في مجال تطبيق النظام الخبير.

٨-١-٣ تطبيقات النظم الخبيرة Expert Systems Application

من أهم مجالات تطبيقات نظم الخبرة هي التصنيف (classification). حيث يكون مطلوباً من النظام تحديد الفئة أو الطبقة (class) التي ينتمي إليها الكائن (object) المطلوب تصنيفه. و لقد تنوعت مجالات تطبيق النظم الخبيرة و أهمها :

- الطب : في تشخيص (تصنيف) الأمراض و وصف العلاج.
- الزراعة : في مجال تصنيف النباتات و الحشرات و كذا أمراض النباتات و كيفية رعاية النبات للوصول إلى أفضل إنتاج.
- التنقيب : في البحث و التنقيب عن النفط و الخامات المعدنية.
- الإلكترونيات : في مجال الكشف عن الأعطال و إصلاحها.
- الحاسبات : في تصميم الشبكات و الكشف عن الأعطال و غيره..
- الجيولوجيا : في التنقيب عن المعادن و النفط.
- الهندسة : في حل المشاكل الهندسية بكافة التخصصات.
- التجارة و الاقتصاد : في الإدارة و التخطيط و أسواق المال.

- القانون : فى بناء نظم خبرة فى قوانين معينة لحل المسائل القانونية.
- الدفاع و الحرب : فى ابتكار استخدام تقنى على لوسائل الدفاع و الحرب الإلكترونية.
- التعليم : فى بناء برامج و وسائل تعليمية على مستوى عال من الذكاء تُعَلِّم و تتصح و تفكر و تقيم.
- الشريعة : فى بناء نظم خبرة فى نواحى صعبة مثل المواريث و الزكاة و غيرها.

٨-٢ تطوير النظم الخبيرة

Expert Systems Development

نتعرّف فى هذا الجزء بإيجاز على مراحل إنشاء النظم الخبير و القائمون على ذلك. و نتعرّف أيضاً على بعض النظم الخبيرة الهامة. و نقدم فى باقى الفصل تطويراً لأحد نظم الخبرة.

٨-٢-١ العاملون فى النظم الخبيرة

Expert Systems Constructors

عند بناء نظام خبير يجب توفر عنصرين هامين للوصول إلى نظام جيد يستطيع أن ينجز عملاً قيماً فى مجال التطبيق المقترح. هذان العنصران لا غنى عن أى منهما و هما :

- مهندس المعرفة (Knowledge Engineer)

هو المبرمج الذى يقوم بتحليل المشكلة وكتابة برامج فى مجال الذكاء الاصطناعى. و ليس بالضرورة أن يكون مهندساً، لكن جرت العادة على استخدام تلك التسمية.

▪ خبير المجال (Domain Expert)

هو الشخص المتخصص في مجال معين وليس بالضرورة أن يكون على درجة كبيرة من العلم أو أن يكون لديه أى دراية بعلم الذكاء الاصطناعى أو حتى أى إلمام بمبادئ الحاسب الآلى. للمهم هو مدى خبرته و إلمامه ببواطن الأمور فى مجال تخصصه.

٨-٢-٢ خطوات إنشاء النظم الخبيرة Expert Systems Steps

إنشاء نظام خبير يمر بعدة خطوات أو مراحل شأنه شأن نظم المعلومات و إن اختلفت فى المفهوم. خطوات إنشاء النظم الخبيرة هى :

▪ تعريف التطبيق

فى هذه المرحلة يتم تحديد خصائص التطبيق و مجال الخبرة بشكل دقيق جداً.

▪ تكوين المفاهيم

فى هذه المرحلة يتم تحليل المشكلة بشكل أعمق ووضع الرسومات والنماذج التى تبين المعرفة المتعلقة بالتطبيق وعلاقتها ببعضها البعض.

▪ تصميم النظام (تحديد قاعدة المعرفة و محرك الاستدلال)

فى هذه المرحلة يتم تحديد الأدوات المستخدمة فى بناء كل من : قاعدة المعرفة (Frame, Semantic Net, Rules, Case_based) و أدوات البحث وطرق الاستدلال على الحلول وكذلك اشتقاق معرفة جديدة.

▪ برمجة النظام

فى هذه المرحلة يتم برمجة التطبيق باختيار لغة برمجة مناسبة للتطبيق.

■ اختبار النظام و توثيقه

آخر مرحلة يتم فيها اختبار النظام باستخدام بيانات حقيقية من قبل أشخاص ليس لهم علاقة بالبرنامج، و ذلك للتحقق من صحة النظام وكفاءة واكتمال التصميم.

فى البداية يتم إنشاء نظام تجريبى (prototype system). ثم يُترك النظام فترة تحت الاختبار بعدها يُعاد بناء النظام من جديد مع علاج كافة المشاكل التى ظهرت أثناء تشغيل النظام التجريبى.

٨-٢-٣ أمثلة من النظم الخبيرة Expert Systems Examples

فى الحقيقة يوجد العديد من نظم الخبرة لا يسعنا المجال لسرد جميعها. فقط نقد بعضاً منها على سبيل المثال لا الحصر :

■ نظام Eliza للعلاج النفسى

هو نظام يقوم بإجراء حوار مع المستخدم ويحاول أن يوجه المستخدم ويستخرج منه المعلومات الضرورية و يوجه الحوار كطبيب نفسى.

■ نظام Mycin الطبى (١٩٧٢)

يطلب النظام من المستخدم إدخال بياناته الشخصية ثم الأعراض المرضية ثم نتائج التحاليل والفحوصات حتى يحصل على تشخيص وعلاج مناسب للأمراض الدم، و لقد سُمي محرك الاستدلال الخاص بهذا النظام باسم Emycin أى Mycin الفارغ، لإمكانية استخدامه بعيداً عن Mycin فى تطبيق آخر.

- **نظام HYPO لقضايا التجارة (١٩٨٧)**
نظام مساعدة لتصنيف الحالات القانونية التجارية و اتخاذ ما يناسبها.
- **نظام PROTOS فى المجال الطبى (١٩٨٩)**
نظام لتصنيف الحالات الطبية و انتقاء أقرب الحالات المُخزَّنة للحالة الجديدة و من ثم تصنيف الحالة الجديدة.
- **نظام ARCHIE لمكتبة معمارية (١٩٩٢)**
يضم النظام مكتبة من حالات التصميم المعماري و يساعد المهندسين المعماريين فى الرسم و التصميم و يقدم التصميم المناسب من مكتبة التصميمات.
- **نظام CASCADE لمساعدة مهندسى الحاسب (١٩٩٢)**
هذا النظام يساعد المهندسين فى حل المشكلات التى تحدث عند تعطل الأجهزة التى تعمل بنظام تشغيل VMS على أجهزة digital.
- **نظام Help-Disk لمستخدمى الحاسب (١٩٩٣)**
نظام مساعدة لحل مشاكل مستخدمى الحاسب. فهو يوفر حلول مشاكل مستخدمى الحاسب فى General Electric.
- **نظام Textile of EMPA لعيوب النسيج (١٩٩٤)**
يُستخدَم هذا النظام لتحديد عيوب النسيج و يعتمد تقنية التصنيف لكشف عيوب الصناعة، و اتخاذ ما يناسبها.

■ نظام Nasser96 للتصنيف (١٩٩٦)

هذا النظام عبارة عن أداة تُستخدم للتصنيف و التشخيص و قد جُرِّبَت في تشخيص أمراض الحساسية و وصف العلاج. ثم جُرِّبَت في تصنيف حشرات المزارع و تحديد طرق الوقاية أو المقاومة. سوف نقدم هذا النظام بشيء من التفصيل في بقية هذا الفصل.

٨-٣ أداة التصنيف و التشخيص : ناصر ٩٦

Classification and Diagnosing Tool : Nasser96

هذا العمل هو نظام يعمل كأداة للاستدلال المنطقي على الحالات في مجال التصنيف (Case_Based Reasoning System using Object_Oriented Techniques). بُنِيَت هذه الأداة باستخدام Visual C++ (ناصر، ١٩٩٥ و ١٩٩٦).

تقوم الأداة باستنباط (جلب) المعرفة ألياً سواء كانت وصفية أو استراتيجية في مجال التطبيق. يستقبل النظام هذه المعرفة من الخبير في صورة حالات، ثم يقوم بالاستدلال على الحل بالبحث في قاعدة المعرفة عن حالات مماثلة بهدف تصنيف أو تشخيص الحالات الجديدة.

حيث تستقطب الأداة الحالة الجديدة المُصنَّفة كنموذج جديد أو دمجها في حالة موجودة من قبل، في حالة موافقة الخبير على التشخيص. يتم ذلك في قاعدة معرفة عبارة عن شبكة تعتمد في تصميمها على الأساليب الفنية للتقنيات مُوجَّهة الأهداف (Object_Oriented Techniques). هذه القاعدة مصممة بحيث يمكن للخبير في أي مجال استخدامها في التصنيف دون أي تعديل يُذكر.

يُمثل النظام الحالات المنتمية إليه، و لا يوجد أى تعريف مُسبق صريح لأصناف أو فئات أو طبقات (classes) معينة فى مجال التطبيق. لكن النظام يشق تعريف الطبقة من الحالات أو الكائنات (Objects) المنتمية إليها. يُمثل الصنف كعقدة أو نقطة (node) متصلة بعلاقات رابطة مع الحالات المنتمية إليها فى شبكة معرفة لفظية (semantic net). كذلك يتم تمثيل الحالة أو الكائن بنقطة أو عقدة ترتبط بخصائصها.

تتميز الأداة ناصر٩٦" بعدة مميزات أبرزها استنباط المعرفة المطلوبة بأنواعها المختلفة، كذلك لا تكتفى بتشخيص أو تصنيف الحالات المُدخلة، لكنها تحدد الأفعال التى يجب إجرائها للحالة المُصنّفة. تم اختبار الأداة فى تطبيقين مختلفين. التطبيق الأول هو تشخيص أمراض الحساسية و تحديد الاختبارات اللازمة للمريض و خطوات العلاج. التطبيق الثانى هو تصنيف حشرات المزارع و طرق مقاومتها ميكانيكية أو بيولوجية أو إشعاعية.

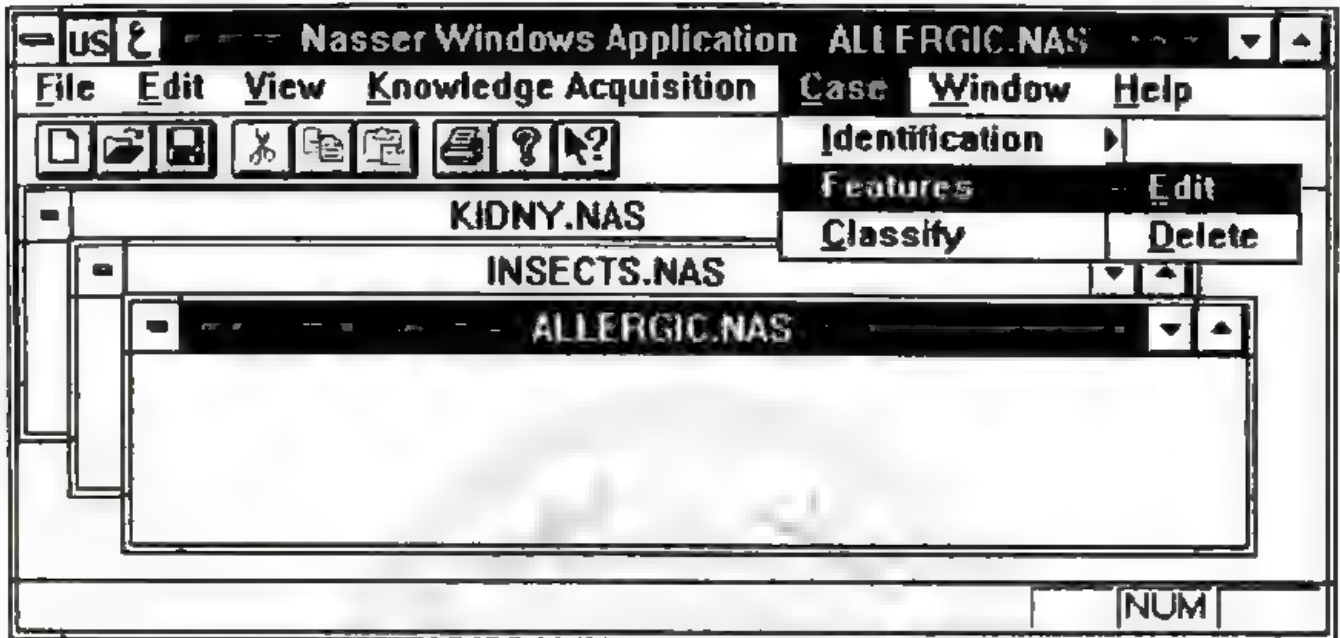
لقد وضع النظام أسلوباً جديداً فى برامج الخبرة و المعرفة ألا و هو إمكانية استخدام النظام لعدة قواعد للمعرفة، كل منها فى مجال تطبيق مختلف، و هذا ما جعله أداة للتصنيف. كذلك فإن النظام قادر على التعلّم و استقراء المعرفة من الخبير فى حال الفشل فى التصنيف كما فى حالة نجاحه و بتوجيه خبير المجال (domain expert).

٨-٤ وصف الأداة التصنيف ناصر٩٦

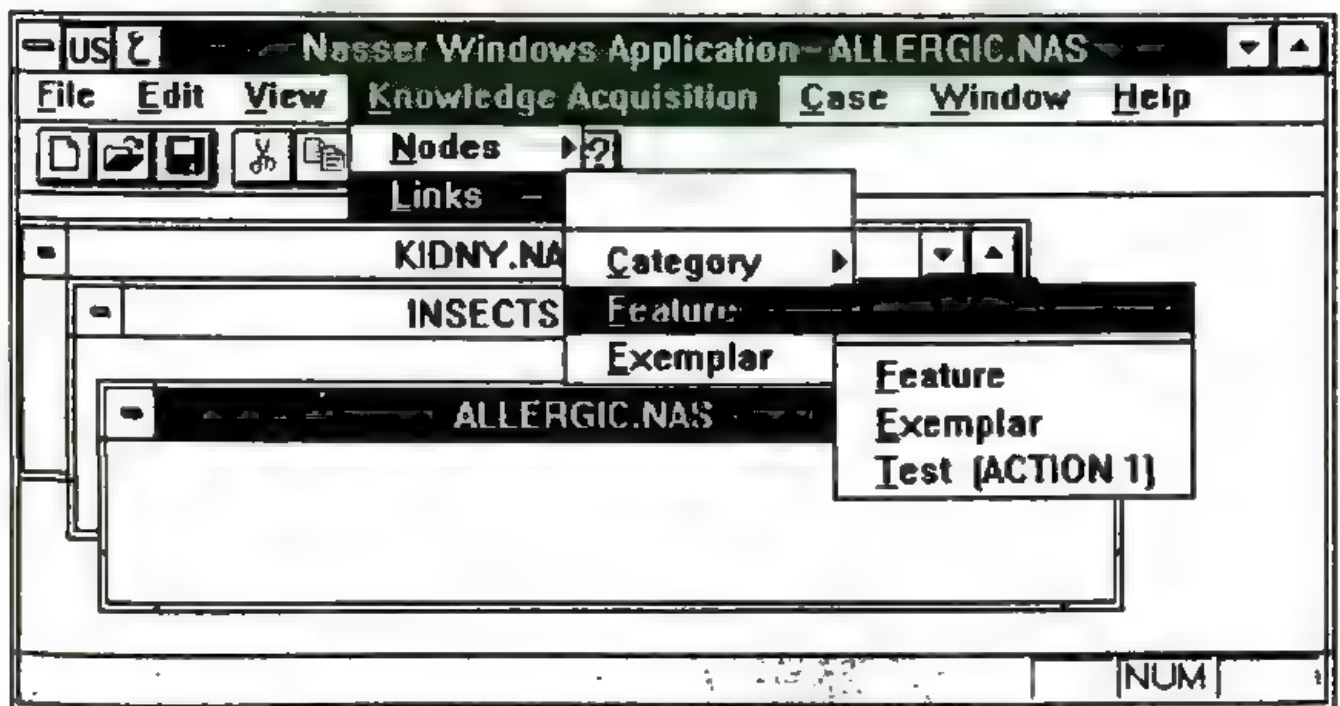
Overview of Nasser96 Tool

توفر أداة التصنيف ناصر٩٦ مساعدة فعالة فى حل المشاكل و تستطيع جلب المعرفة الجديدة فى مجال التطبيق من خلال حالات و أمثلة عامة للتدريب. فهى تقوم بالتصنيف تحت إشراف الخبير و لا يعمل مهندس المعرفة كوسيط بين الخبير

و الأداة. شكل ٨-١ يعرض واجهة الأداة مع القائمة الخاصة بالتصنيف. بينما يعرض شكل ٨-٢ واجهة الأداة مع قائمة التحكم المباشر للخبير في قاعدة المعرفة.



شكل ٨-١ : واجهة الأداة مع القائمة الخاصة بالتصنيف.

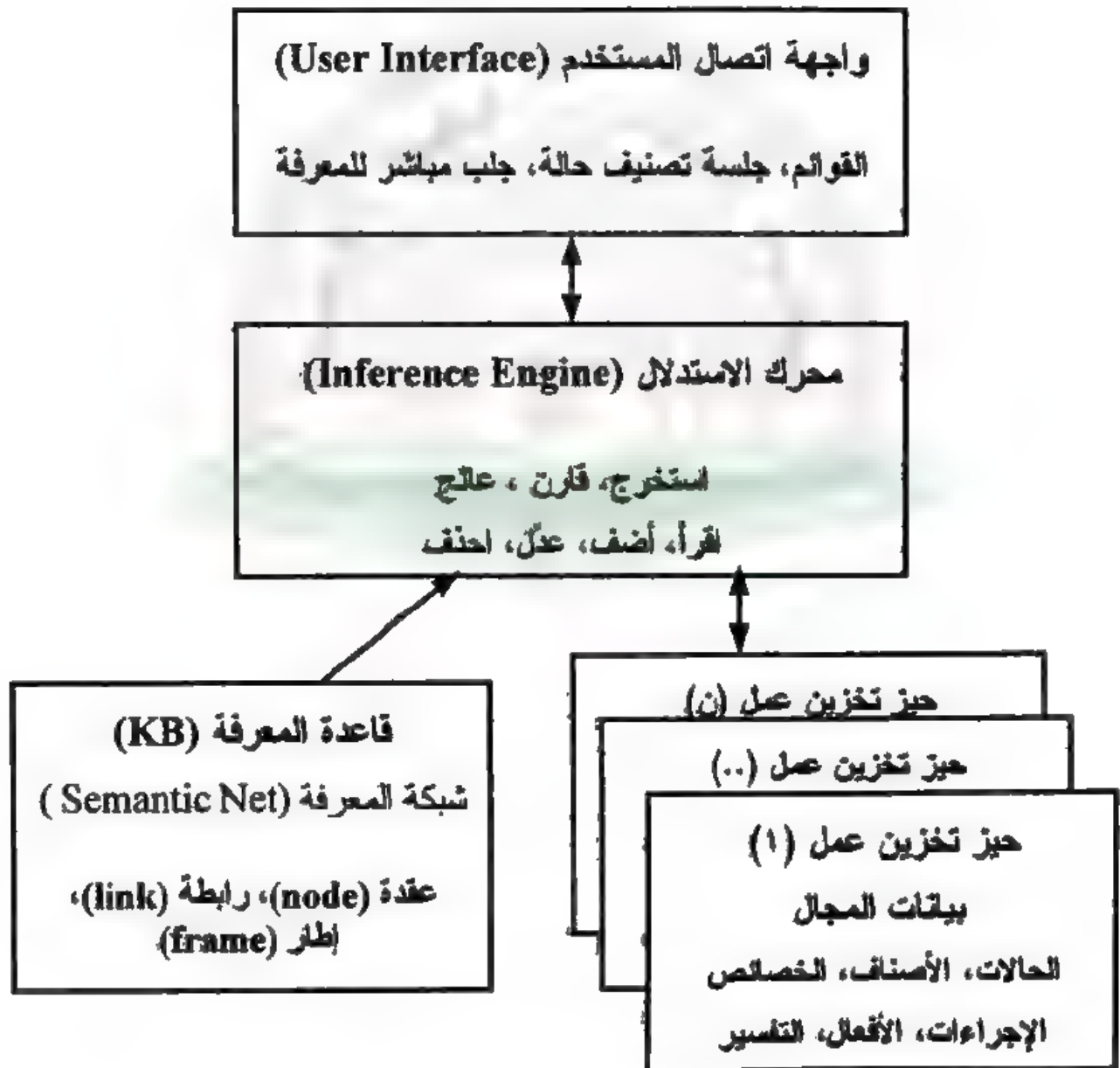


شكل ٨-٢ : واجهة التطبيق مع قائمة التحكم المباشر في قاعدة المعرفة.

٨-٤-١ تركيب أداة التصنيف

Structure of the Proposed Tool

نظراً لأن الأداة ستقوم بالتشخيص و التصنيف لحالات مُنخَلة، فقد صُمِّمَت مكوناتها لتحقيق هذا الهدف. و كانت النتيجة خروج أداة قادرة على التصنيف في العديد من المجالات. تتكوّن الأداة من أربعة أجزاء أساسية، تتعاون معاً لتُثبِت عمومية الأداة. كما و أن استخدام التقنيات الموجهة الأهداف المستخدمة في تحليل مجال التطبيق أدت أيضاً إلى عمومية الأداء. شكل ٨-٣ يعرض تركيب الأداة.



شكل ٨-٣ : تركيب أداة ناصر ٩٦.

■ واجهة اتصال المستخدم (User Interface)

تتكون واجهة اتصال المستخدم بالأداة من حوارات موجهة بالقوائم لتساعد المستخدم على التفاعل مع الأداة مباشرة بسهولة ويسر. يستطيع المستخدم تمثيل الحالات في جلسات عمل مع الأداة.

■ محرك الاستدلال ((Inference Engine (Reasoner))

يستدل محرك الاستدلال في الأداة باستخدام تقنية التسلسل العكسي (backward chaining) ليلتقط الاختيار الأفضل من عدد كبير من الحلول الممكنة. ويتعامل مع الحالات و الكائنات الأخرى و العلاقات الرابطة بينهم بطريقة قياسية في ظل وجود عد دقة و غموض باستخدام نظرية المجموعة الغامضة (Fuzzy set theory).

■ قاعدة المعرفة ((Knowledge Base (KB))

قاعدة المعرفة في هذه الأداة هي شبكة معرفة لفظية (Semantic Net) تتكون من نقاط/عقد (nodes) و روابط (links). تركيب هذه القاعدة مبنى على التحليل الموجه الأهداف لمجال التطبيق، لتوزيع الحالات (cases) التي تحتفظ بها الأداة. لذا يُطلق عليها case_based knowledge و تضم تركيب مُمَهَّزَس يستخدمه محرك الاستدلال للعثور على الحالات. و نستطيع قاعدة المعرفة أن تتكيف مع التغييرات في المجال أو حتى تغيير المجال نفسه.

■ حيز تخزين العمل (Working Storage)

تستطيع الأداة إنشاء أكثر من حيز تخزين العمل. و يكون حيز تخزين العمل منفصلاً عن قاعدة المعرفة للأداة. كل حيز تخزين عمل يُخصَّص لمجال ما، و عندما يكون فعالاً تكون الأداة متخصصة في هذا المجال فقط.

٨-٤-٢ وصف الحالة المُدخلة

Description of a Presented Case

الحالة (case) هي الكائن المطلوب تصنيفه و ربطه بطبقة (class) أو صنف (category). و تتكون الحالة من مجموعة خصائص (features) و قيمهم. سوف نعرض حالة مريض تم إدخالها للنظام فى بداية تشغيله لاحتوائها على كمية كبيرة من المعلومات. الجدير بالذكر أن كمية المعلومات المصاحبة للحالات التى يصنفها النظام تتناقص مع تزايد خبرة النظام.

شكل ٨-٤ يعرض نموذج لحالة مريض تتضمن البيانات الشخصية و الأعراض المرضية. بينما يعرض شكل ٨-٥ العلاقة بين خصائص الحالة و الطبقة التى تم اختيارها لتصنيف الحالة. يعرض شكل ٨-٦ عائلة (شجرة طبقات) الأصناف للصنف الذى انتمت إليه الحالة.

CASE بيانات تحديد الحالة			
Case Id. : Sayfo Allah		Area : Cairo	
Date : 1/10/1996		Phone : 747120	
Case Features			
خصائص الحالة			
Fno.	Feature Name اسم الخاصية	Value1 قيمة	Value2 قيمة
F1	wheezing		
F2	cough	early morning	night
F3	dyspnoea		
F4	night symptoms	> 2/month	
F5	Peak Expiration Flow	>=80%	
...			
Fn			

شكل ٨-٤ : وصف حالة مريض.

FNo.	Qualifier		Relation Name	
	Text	Number	Text	Number
F1	always	1.0	resulted from	1.0
F2	usually	0.8	resulted from	
F3	somtimes	0.7	imply	0.7
F4	usually		suggest	0.8
F5	usually		suggest	
...				
Fn				

شكل ٨-٥ : العلاقة بين خصائص الحالة و تصنيف الحالة.

Case Classification تصنيف الحالة	Sub-Categories الأصناف الفرعية (الأبناء)
Intermittent Asthma	...
Super Category الصنف الأعلى (الأب)	
Asthma	

شكل ٨-٦ : عائلة الطبقة (الصنف).

يعرض شكل ٨-٧ الفحوص و الاختبارات التي أجريت للحالة (الإجراء الأول). يعرض شكل ٨-٨ الإجراءات المنفذة (الإجراء الثاني). أخيراً، يعرض شكل ٨-٩ علاقة الخصائص ببعضها البعض.

TNo.	Test Name	Method	Feature
T1	Peak Expiration Flow		F1
Tn			

شكل ٨-٧ : الفحوص و الاختبارات التي أجريت للحالة (الإجراء الأول).

PNo.	Procedure Name	Method	Feature
P1	inhaled short acting B2 agonist	when needed	F2
Pn			

شكل ٨-٨ : الإجراءات المُنفَّذة (الإجراء الثاني).

FNo	Qualifier		Relation Name		Neighbor Feature
	Text	Number	Text	Number	
F1	always	1.0	equivalent	1.0	PFE
...					
Fn					

شكل ٩-٨ : علاقة الخصائص ببعضها.

٨-٤-٣ الحدود و القيود فى أداة ناصر ٩٦

Limitations in Nasser96 Tool

الجدير بالذكر أنه لا يوجد قيد على عدد الحالات أو خصائص الحالة أو الاختبارات أو الإجراءات سواء من ناحية العدد أو المسميات. تتكئف الأداة مع المعرفة الحقيقة الموجودة فى مجال التطبيق.

مثل معظم النظم الخبيرة، توجد بعض القيود فى أداة التصنيف ناصر ٩٦. هذه القيود تؤثر فى قدرة الأداة و خصوصاً على المدى البعيد. فيما يلى نسرّد بعضاً من هذه الحدود و القيود :

- فى المجالات الطبية (أو المجالات الشبيهة)، من الممكن أن يكون لدى الحالة (المريض) شكوى و أعراض لأكثر من مرض فى نفس الوقت. فى

حين يقتصر عمل الأداة إلى التصنيف إلى مرض أو صنف واحد فقط في نفس الوقت.

■ كل خاصية تستخدمها الحالة في وصف الحالة لها معاملين اثنين فقط وربما يكون هناك حاجة لمعاملات أكثر. لكن بتحليل مجالين تم تطبيق الأداة خالهما لم نحتاج إلى أكثر من معاملين للخاصية.

■ معاملات الخاصية تعالجها الأداة كمتغيرات نصية حرفية و لا مجال لاستخدام الأرقام. حتى لو كتبنا أرقام فلا مجال للمقارنة أو الحسابات على المعاملات. لكن بتحليل مجالين تم تطبيق الأداة خالهما لم نحتاج إلى معاملات رقمية.

■ الخصائص التي تصف الحالة عبارة عن معرفة خبرية (declarative knowledge) كقياسات و لا مجال للمتغيرات الزمنية. و لقد تم ذلك لأن الأداة تُستخدَم في تشخيص و تصنيف حالات لا تحتوى على معاملات زمنية.

■ لا تقوم الأداة بعمل تحسين أو تصفية (refinement) على المعرفة التي جلبتها الأداة من الخبير. أى أنه لا يستطيع أن يحدد أو يعالج أخطاء الخبير. هذا لأن الأداة تعتمد على اكتساب الخبرة من الخبير و القرار النهائي في يد الخبير.

٨-٥ المعرفة المُمَثَّلَة في الأداة ناصر ٩٦

Knowledge Represented by NASSER96

تجلب أداة التصنيف أغلب خبرتها و معرفتها من الحالات المُدخَّلة لتصنيفها. لذلك يُطلق عليها قاعدة المعرفة المبنية على الحالات (case_based knowledge base). و هي تمثل مجال التطبيق -الذي يضم أنواعاً مختلفة من المعرفة خبرية و إجرائية و إستراتيجية- في شبكة معرفة لفظية (semantic net) ذات تسلسل هرمي.

٨-٥-١ تركيب شبكة المعرفة اللفظية

Semantic Net Structure

يعتمد تركيب شبكة المعرفة على التحليل المبني على التقنيات الموجهة (Object-Oriented Analysis). بناءً على ذلك يتم تصنيف المعرفة الموجودة في أي مجال تطبيق إلى كائنات (objects) منتمة إلى فئات/طبقات (classes) و العلاقات الرابطة بين كل منهم.

الطبقات و كائناتها مثل الحالات و خصائصها و الأصناف و الاختبارات و الإجراءات يتم تمثيلها في نقاط/عقد (nodes) الشبكة و يجرى ملؤها بالمعرفة الضرورية الموجزة. أما العلاقات أو الروابط (links) بين الكائنات و بعضها البعض فتحتوى على أغلب المعرفة في مجال التطبيق.

تُستطيع الأداة إنشاء ستة أنواع من النقاط/العقد لتمثيل كائنات المجال الموجودة في التطبيق. تضم العقد كلاً من : الأصناف و الحالات المُدخَّلة للتصنيف و الحالات التي استقطبها النظام و الخصائص و الاختبارات و الإجراءات.

كذلك تستطيع الأداة إنشاء عشرة أنواع من الروابط لتمثيل العلاقات بين العقد الستة السابق ذكرها. هذه الروابط تمثل نظام فهرسة فعال لاستخراج المعرفة المطلوبة. تضم الروابط كلاً من : صنف-صنف و حالة مُصنَّفة-و اختبار و حالة جديدة-حالة مُصنَّفة و حالة مُصنَّفة-صنف و خاصية-صنف و خاصية-حالة جديدة و خاصية-حالة مُصنَّفة و خاصية-خاصية و خاصية-اختبار و حالة مُصنَّفة-إجراء.

من المهم أن نذكر أن عقدة واحدة يمكن أن ترتبط من خلال العديد من الروابط بعقدة أخرى في الشبكة. يحدث هذا في حالة وجود أكثر من علاقة رابطة بين كائنين. كل علاقة رابطة تضم و/أو تعتمد على عناصر معينة مُخصصة لأحد الكائنين أو كلاهما. كلا من العقد و الروابط يتم تمثيلها باستخدام إطارات المعرفة.

٨-٥-٢ جلب و تمثيل التفسير

Explanation Learning and Representation

عندما يقدّم الخبير حالات جديدة للتصنيف، تحاول الأداة أن تصنّف الحالات و أن تفسّر هذا التصنيف. و إذا كان التصنيف غير سليم أو التفسير غير كافٍ، يُطلب من الخبير المُعلّم أن يمد الأداة بالتفسير الملائم و بمعلومات إضافية.

تكتسب الأداة التفسيرات الخاصة بعلاقتين في موضعين. الأول، عندما تتم إضافة حالة جديدة إلى قاعدة المعرفة أو تُدمج في حالة قديمة مُصنَّفة، تكون الأداة قادرة على صياغة تفسير وجود علاقة بين كل خاصية للحالة و بين التصنيف الذي انتمت إليه.

تبحث الأداة عن رابطة بين كل خاصية و التصنيف. فإذا فشلت في ذلك فإنها تطلب مساعدة الخبير في تفسير العلاقة. فهي تسأل عن تفسير لماذا يزيد وجود خاصية معينة من الاعتقاد بأن التصنيف سليم.

يتم تحليل التفسير لتقدير نوع و طبيعة العلاقة. فهناك نوعين من العلاقة الرابطة بين الخاصية و الصنف. الأولى علاقة مُرْشَحة (reminded relation) و الثانية علاقة رافضة عكسية (rejection relation). بعد ذلك تحسب الأداة أهمية العلاقة الرابطة بين حالة مُصنَّفة و بين الصنف. تقدير التمثيل يسمح بتخصيص قيمة رقمية لأهمية خصائص الحالة المُصنَّفة.

العلاقات المُرْشَحة أو الرافضة و أهمية الخصائص و العلاقة الرابطة بين الحالة المُصنَّفة و الصنف تمثل فهارس هامة تستخدمها الآلة في تمثيل المعرفة و استخراج المعرفة المطلوبة.

٨-٦ التصنيف و الاستدلال على الحل باستخدام الحالات

Classification & Case-Based Reasoning

تبدأ الأداة عملية التصنيف لحالة جديدة ببناء قائمة من الأصناف المُرْجَحة و التي ترشحها الأداة بناء على المعرفة المتوفرة مع الحالة و أول صنف في القائمة هو الأكثر ترجيحاً ليكون هو التصنيف. ثم تستخرج الحالات المُصنَّفة من قبل و منتمية إلى التصنيف المقترح و موجودة في قاعدة المعرفة. تبدأ الأداة بالحالة الأكثر تمثيلاً للتصنيف المقترح. فإذا فشلت العملية في استخراج الحالة المناسبة فإن الأداة تعود أدراجها لتستخرج حالة مُصنَّفة أخرى.

Reasoning Process

٨-٦-١ الاستدلال على الحل

عندما يقدم خبير حالة إلى الأداة لتصنيفها، فإنها تتجزئ عملية الاستدلال على الحل خلال عدة مراحل. أغلب الخطوات تدمج استنباط المعرفة و التعلم في التصنيف الموجه للكشف (heuristic classification). شكل ٨-١٠ يعرض مراحل عملية الاستدلال على الحل.

خطوات خوارزم الاستدلال على الحل :

- يبدأ برنامج الاستدلال على الحل عمله بالبحث عن الفحوص و الاختبارات التي يجب إجرائها للحالة المطلوب تصنيفها، قبل الشروع في عملية التصنيف. تعرض الأداة خبرتها و معرفتها السابقة على الخبير و تطلب منه إجراء بعض الاختبارات أو الفحوص و مدها بالنتائج مع خصائص الحالة و تخيره في البدء في التصنيف دون انتظار نتائج الفحوص.
- بعد إنهاء الجزء السابق، يبدأ البرنامج بناء قائمة من الطبقات أو الأصناف التي يتوقع أن يكون التصنيف (الحل) من بينها.
- فإذا كانت هذه القائمة خالية من أي تصنيف لعدم وجود خبرة سابقة، تبدأ الأداة تنفيذ الدالة الخاصة بقيام الخبير بالتصنيف و تطلبه منه. كذلك تطلب الأداة تفسيرات من الخبير عن هذا التصنيف و ارتباطه بخصائص الحالة.
- إذا احتوت القائمة على عدة تصنيفات و رفضها الخبير جميعاً، فإن الأداة تطلب مدها بالتصنيف و تطلب منه تفسير رفضه لمقترحاتها لعدم تكرار ذلك فيما بعد.



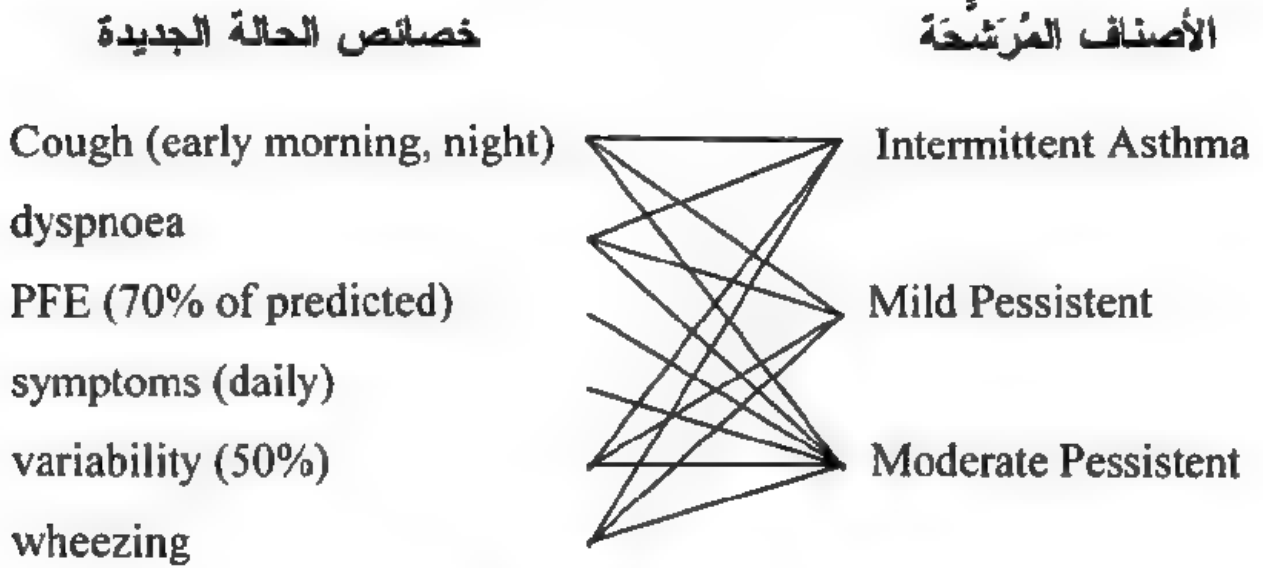
شكل ٨-١٠ خوارزم عملية الاستدلال على الحل.

- أما إذا وافق الخبير على مقترحات الأداة، فإن الأداة تبدأ عملية التصنيف تحت إشراف الخبير. و تبدأ الأداة تشغيل الدالة الخاصة بقيام الأداة بالتصنيف للعثور على الحل.
- يقوم برنامج الاستدلال باستخراج حالة تم تصنيفها من قبل و يستخدمها كنموذج للمطابقة مع الحالة الجديدة. فإذا كانت المطابق غير صحيحة فإنه يستخرج حالة مُصنَّفة أخرى منتمية لنفس التصنيف أو التصنيف التالي في قائمة الأصناف المرشحة.
- تستمر العملية إلى أن يعثر برنامج الاستدلال على المطابقة السليمة أو يفشل في ذلك. فإذا نجح يطلب من الخبير الموافقة النهائية على التصنيف. بينما في حالة الفشل، يطلب من الخبير أن يقوم بالتصنيف بنفسه و يطلب منه التفسيرات الضرورية.
- بعد الموافقة على تصنيف معين بشكل نهائى أو الحصول عليه من الخبير، يبدأ برنامج الاستدلال البحث عن الإجراءات الواجب تنفيذها مع الحالة لمُصنَّفة (طريقة علاج المريض أو طريقة مقاومة الحشرة مثلاً) من خبرته السابقة. يعرض البرنامج ما لديه ويطلب من الخبير الموافقة و مدّه بالجديد إن لزم ذلك.

Classification

٨-٦-٢ التصنيف

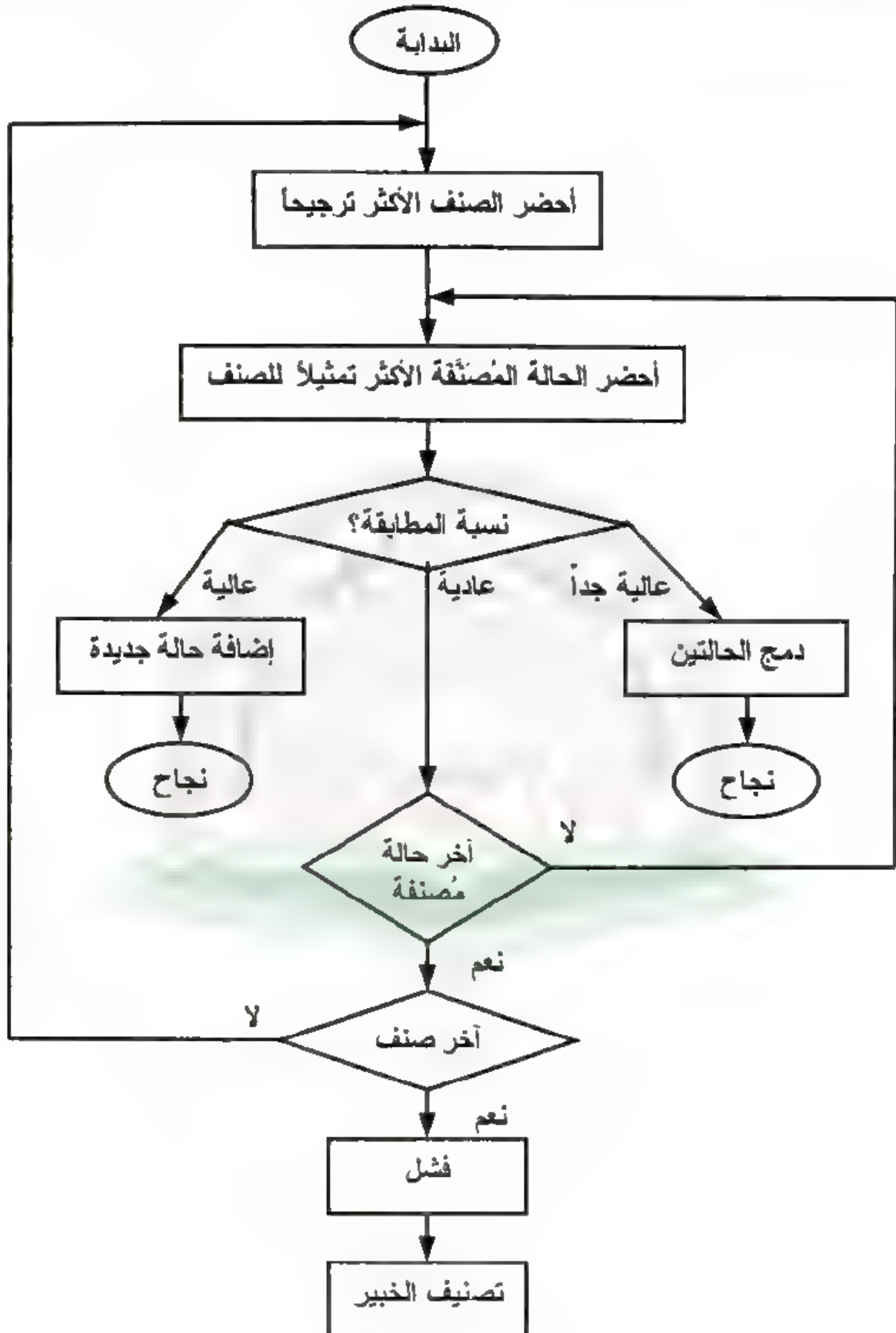
عندما يقبل الخبير أن تستمر عملية التصنيف من خلال قائمة التصنيفات المرشحة من قبل النظام. شكل ٨-١١ يوضح كيف تُرشح خصائص الحالة قائمة من الأصناف. شكل ٨-١٢ يعرض خوارزم التصنيف الذى تستخدمه الأداة.



شكل ٨-١١ : رسم توضيحي لترشيح خصائص الحالة الجديدة لمجموعة الأصناف المرشحة.

خطوات خوارزم تصنيف الأداة لحالة جديدة :

- تقارن الأداة الحالة الجديدة بالحالات المُصنَّفة من قبل و التي تنتمي إلى الأصناف المرشحة. و تبدأ بالحالات المنتمية إلى الطبقة أو الصنف الأكثر ترجيحاً و المدعوم من خصائص الحالة الجديدة.
- داخل الصنف يستخرج برنامج التصنيف حالة تلو أخرى من الحالات الموجودة بقاعدة المعرفة و يحاول مطابقتها مع الحالة الجديدة و يبدأ بحالة قديمة ذات التمثيل الأكبر للصنف.
- يتم إجراء عملية المطابقة باستخدام عملية مقارنة النموذج المبنية على المعرفة (knowledge-based pattern matching process). و تستمر هذه العملية حتى العثور على حالة قديمة مشابهة للحالة الجديدة و يوافق عليها الخبير أو تفشل الأداة في العثور على حالة مشابهة.



شكل ٨-١٢ : خوارزم تصنيف الأداة للحالات.

- إذا فشلت الأداة، يقوم برنامج الاستدلال على الحل في تفعيل دالة تصنيف الخبير ليقيم الخبير بمهمة التصنيف كما قدّمنا من قبل.

٧-٨ تطبيقات الأداة Tool Applications

تم تطبيق أداة التصنيف ناصر ٩٦ في مجالين مختلفين. المجال الأول مجال طبي و هو تشخيص أمراض الحساسية و تحديد خطة علاجها. المجال الثانى هو تصنيف حشرات المزارع و تحديد طريقة مقاومتها.

١-٧-٨ تشخيص أمراض الحساسية

Diagnosing Allergic Diseases

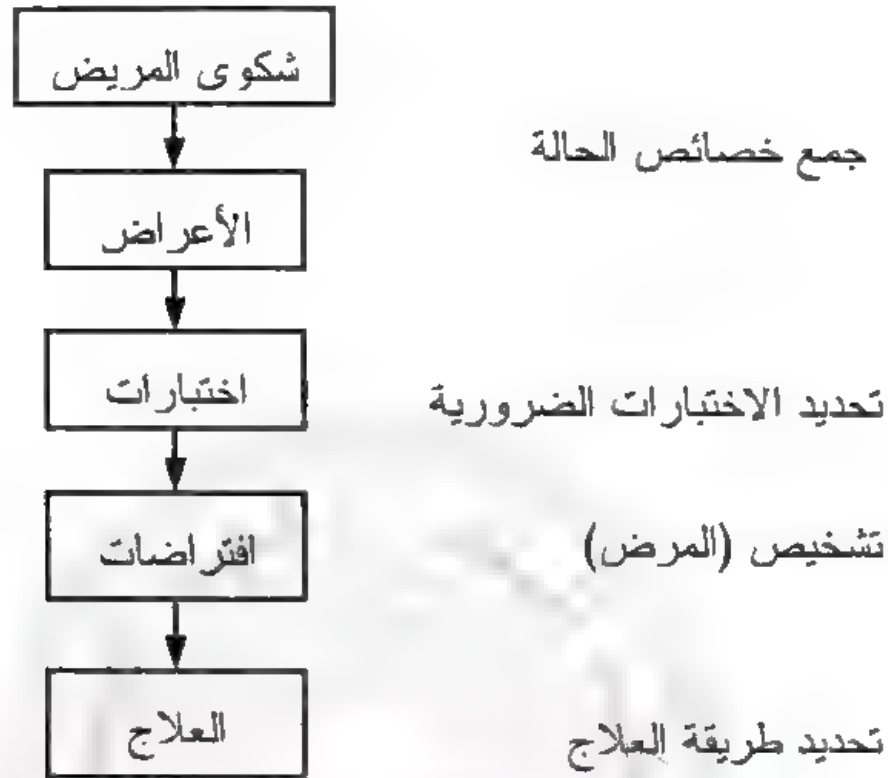
مهمة هذا التطبيق هى إدارة عمل طبي و هو عملية تشخيص المرضى و تصنيف الأمراض و تحديد العلاج. من الواضح أن الخبير الطبى يستخدم مجموعة من الأعراض (symptoms)، و معها عدد محدد من القيم، لتشخيص حالة المريض.

كل خاصية أو صفة أو عَرَض لمرضٍ ما يمكن تقدير قيم محتملة لها تجمع كافة القيم الممكنة لخاصيةٍ ما يحدد كافة حالات المريض المحتملة. عملية التشخيص أى تصنيف حالة المريض هى تحديد المرض المصاب به.

يبدأ الطبيب عمله (ستؤدى الأداة بعملها بنفس الطريقة) بسؤال المريض على شكواه و ملاحظة الأعراض المرضية كما نرى فى شكل ٨-١٣ الذى يوضح طريقة تشخيص الطبيب و الأداة لحالة مريض.

ثم يقوم الخبير بوضع أكثر من تخمين أو افتراض ظنى (hypothesis) للمرض و ربما يطلب من المريض إجراء بعض الفحوصات و التحاليل و إحضار النتائج لجمع مزيد من المعلومات. بعد ذلك يختار أحد هذه الافتراضات (الأرجح) و

هذا هو التشخيص أو التصنيف. بناءً على التصنيف يحدد الطبيب طريقة العلاج (therapy) و التداوى (treatment).



شكل ٨-١٣ : عملية تشخيص الأمراض.

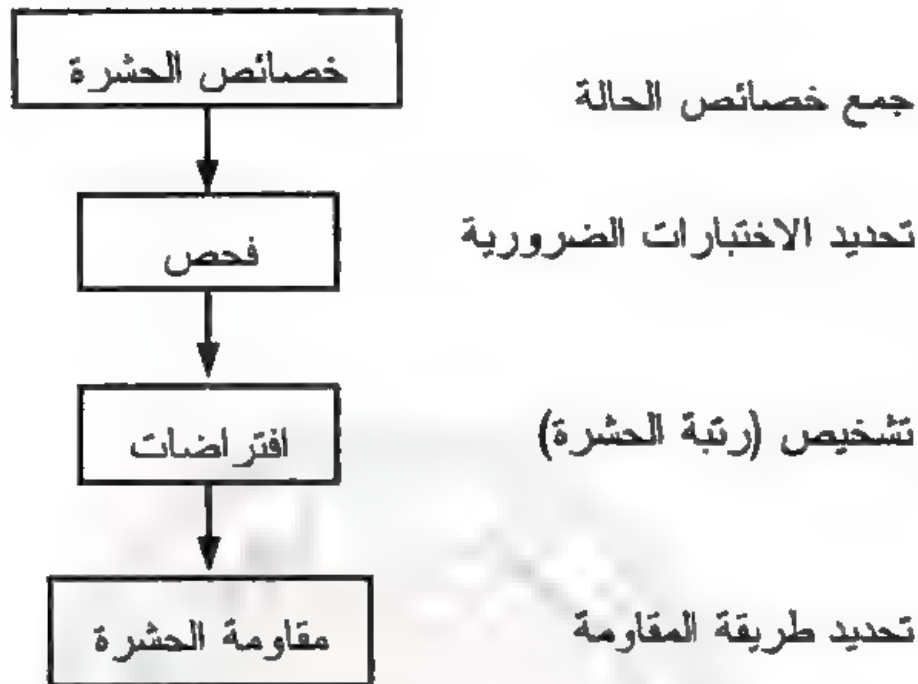
نظراً لاستخدام الخبير لألفاظ معينة لوصف الحالة المرضية فإن الأداة تستخدم طريقة المنطق الغامض (fuzzy logic) و نظرية المجموعة الغامضة (fuzzy set theory) اللذين قدّمناهم في الفصل السابق.

٨-٧-٢ تصنيف حشرات المزارع

Classification of Farm's Insects

مهمة هذا التطبيق هي تصنيف الحشرات الموجودة في المزارع و تحديد رتبة أو عائلة الحشرة (نوعها). بعد ذلك يتم تحديد طريقة المقاومة المناسبة سواء كانت

يدوية أو ميكانيكية أو بيولوجية. شكل ٨-١٣ يوضح طريقة عمل الخبير الزراعى و الأداة فى التصنيف و تحديد طريقة المقاومة للحشرة المُدخلة.



شكل ٨-١٣ : عملية تصنيف حشرة المزارع.

يستخدم الخبير الزراعى مجموعة من الخصائص (كل منها له عدد محدد من القيم) للتعرف على الحشرة و تحديد رتبتها. تجمع الخصائص لحشرة ما يؤدي إلى التعرف على الحشرة. يتحرى الخبير خصائص و مواصفات الحشرة (الحالة). و بعد جمع خصائص الحشرة ربما يتفحص الحشرة مجهرياً لجمع المزيد من الخصائص. ثم يقترح تصنيفاً معيناً مُرجّحاً من بين عدة افتراضات محتملة.

نظراً لاستخدام الخبير الزراعى لأوصاف دقيقة فى وصف الحشرة فإن الأداة لا تستخدم طريقة المنطق الغامض (fuzzy logic) و نظرية المجموعة الغامضة (fuzzy set theory).

٨-٨ أسئلة

١. عرّف كلاً من :

Expert system, domain expert, knowledge engineer

٢. ما هي مميزات النظم الخبيرة؟ و ما هي مشاكلها؟

٣. لماذا نحتاج إلى إنشاء نظم خبيرة؟ و في أى مجالات؟

٤. من هم العاملون في مجال النظم الخبيرة؟ و ما دور كل منهم؟

٥. وضح خطوات إنشاء النظم الخبيرة.

الفصل التاسع

الشبكات العصبية

Neural Networks

لقد تعلم الإنسان من خلقته التي خلقه الله عليها . فالشبكات العصبية في عقل الإنسان وفي جسمه تعمل بطريقة غاية في الدقة والتعقيد. حيث أن المعلومات لا تخزن في دماغ الإنسان في موقع معين بل تتوزع على خلايا عصبية عديدة وعندما يحاول المرء تذكر أمر ما فإنه يجمع هذه المعلومات من شتاتها. و استطاع الإنسان أن يحاكي ذلك باستعمال شبكات عصبية اصطناعية داخل برامج الحاسب تقوم بحل مسائل عديدة خاصة ما يحتاج منها إلى تمرين وتعلم واستفادة من الماضي.

Introduction

٩-١ مقدمة

أهم مميزات العقل البشري - و التي لا تقتصر على كونها شبكات عصبية هائلة - هي التعلم و الفهم و التفكير، والسؤال هو ما هذا السحر الذي يحدث داخل هذا العقل لكي تحدث عمليات التعلم و الفهم و التفكير؟!

٩-١-١ مفهوم الشبكة العصبية البشرية

Human Neural Network Concept

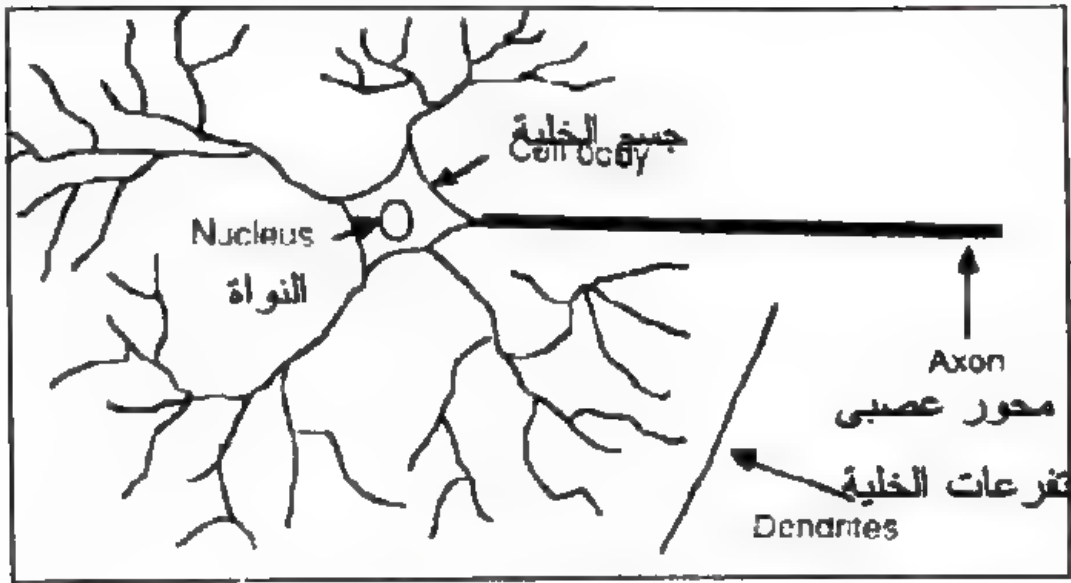
يرى العلماء أن أى إدخال يكون نشطاً عندما تتلقى الخلية العصبية إشارة من الخلايا الأخرى حيث يقوى وتصبح الخلية أكثر حساسية لأي إشارات أخرى قادمة

فى المستقبل. هذه العملية "التي لا يوجد تصور لكيفية حدوثها" هى التى تؤدى إلى حدوث عملية التعلم.

فى عام ١٩٤٩ اكتشف العالم النفسى دونالد هيب اكتشافاً كبيراً حيث قدم مبدأ عام للتعلم، بسيط جداً ولكنه للأسف غير معروف للكثيرين حيث توصل إلى أن توصيلات إدخال المعلومات إلى الخلية العصبية متصلة بتوصيلات خروج البيانات من الخلايا العصبية الأخرى فى الشبكة العصبية. وتقوم هذه الخلية بإرسال الإشارة التى تحفز الخلايا الأخرى عندما يتجاوز إجمالى المداخل قيمة الحد الأدنى. وتأخذ الإشارة التى ترسلها الخلية العصبية شكل انفجار عشوائى من النبضات، ويؤدى ذلك إلى سلوك معقد للشبكة برمتها، وهو سلوك معقد ومن الصعب تحليله. وحتى اليوم ما زلنا نحتاج إلى محاكاة شبكات صغيرة جداً لاكتشاف ما يمكن أن يكون عليه سلوكها.

يُطلق على الخلية العصبية الخلية المُدركة لأن الخلية العصبية عبارة عن إشارات داخلية تدركها الخلية و تُخرج النتيجة المناسبة للإشارة المُدخلة. و تتعلم الخلية بالتدريب و بالطبع لا يمكن تدريبها على كل شىء.

لا يوجد لدينا وصفة محددة للعقل لأن عقولنا تستخدم شبكات أعمق من ذلك، كما و أنه من المحتمل أن تكون عقولنا عبارة عن شبكات من الخلايا المتصلة ببعضها البعض. و تقدم تقريراً لتقييم الموقف بين الخلايا المتصلة ببعضها البعض. وتحديد المواضيع التى تصبح فيه الاتصالات مفيدة مشكلة عويصة. فنوع الشبكات الذى نستطيع تدريبه باستخدام التفاعل العكسى لا يمكن أن يكون به نوع من التقارير المنعكسة و فى الحقيقة لا تحتاج إلى مثل هذه التقارير. شكل ٩-١ يعرض تركيب الخلية العصبية فى عقل الإنسان.



شكل ٩-١ : مكونات الخلية العصبية.

٩-١-٢ العقل البشري نظام معالجة معلومات

Brain as an Information Processing System

يتكوّن العقل البشري من بلايين الخلايا العصبية (neurons). ترتبط كل خلية عصبية بآلاف الخلايا العصبية الأخرى من خلال نقاط اشتباك عصبى (synapses). شبكة الخلايا العصبية الموجودة فى العقل نظام معالجة متوازية للمعلومات.

تأمل الوقت اللازم لكل عملية أولية : الخلية العصبية تعمل بمعدل أقصاه ١٠٠ هرتز بينما يعمل الحاسب الآلى عدة مئات الملايين من العمليات على مستوى الآلة كل ثانية. و مع ذلك يتميز الجهاز العصبى (nervous systems) بالقدرات التالية و التى تسعى إلى نمذجتها :

- معالجة متوازية موزعة للمعلومات.
- درجة عالية من الترابط بين الوحدات الأساسية.
- الاتصال ممكن تعديله بناءً على الخبرة.

- التعلُّم عملية ثابتة و غير مراقبة.
- التعلُّم مبنى على معلومات محلية.
- الأداء يتناقص برشاقة إذا أزيلت بعض الوحدات.

يحاول أحد فروع الذكاء الاصطناعي و هو الشبكات العصبية (Neural Networks) أن يُقَرِّب المسافة بين الحاسبات و قدرات عقل الإنسان بمحاكاة سمات معينة من معالجة المعلومات في العقل بطريقة مبسطة.

٩-١-٣ مفهوم الشبكات العصبية الاصطناعية

Artificial Neural Network Concept

شبكات الخلايا العصبية الاصطناعية (Artificial Neural Network(ANN)) هي نموذج معالجة معلومات مُستوحى من طرق النظم العصبية الحيوية، مثل العقل البشرى فى معالجة المعلومات. العنصر الأساسى فى هذا النموذج هو التركيب الجديد لنظام معالجة المعلومات. فهو يتكوّن من عدد كبير من عناصر المعالجة عالية الاتصال فيما بينها (الخلايا العصبية neurons) تعمل فى تنسيق لحل مشاكل معينة. تتعلم ANNs من الأمثلة مثل البشر.

قد يبدو أن مجال الشبكة العصبية جديد، مع أنه بدأ قبل ظهور الحاسب الآلى و بقى حياً و متوقفاً لفترةٍ ما. الكثير من التقدم تعززت باستخدام محاكاة الحاسب غير الغالية. و بعد فترة حماس، نجا المجال من فترة إحباط و فقدان السمعة الجيدة. خلال هذه الفترة، عندما كان التمويل و الدعم العلمى محدوداً، جاء بعض التقدم على يد عدد محدود من الباحثين.

يتم تشكيل ANN من أجل تطبيق معين، مثل ملاحظة النموذج (pattern recognition) أو تصنيف البيانات (data classification) خلال عملية تعلُّم

(learning process). عملية التعلم فى النظم الحيوية تتضمن ضبط ارتباطات نقاط الاشتباك العصبية الموجودة بين الخلايا العصبية.

أما الآن يتمتع مجال الشبكات العصبية بمزيد من الاهتمام و ما يتبعه من زيادة التمويل. أول خلية عصبية إصطناعية أنتجها كل من عالم الخلايا العصبية البيولوجية Warren McCulloch و عالم المنطق Walter Pitts عام ١٩٤٣. لكن التكنولوجيا المتاحة فى ذلك الوقت لم تسمح لهم بفعل الكثير.

نظراً لأن الشبكات العصبية ذات قدرات ملحوظة على اشتقاق المعنى من البيانات المعقدة جداً أو غير الدقيقة، يمكن استخدامها لاستخلاص النماذج و اكتشاف النزعات الصعب جداً ملاحظتها من قبل البشر أو تقنيات الحاسب الأخرى. يمكن اعتبار شبكة خلية عصبية مُدرّبة كخبير فى نوع المعلومات المُعطاة للتحليل. بعد ذلك، يمكن استخدام هذا الخبير لتوفير إسقاطات تعطى مواقف جديدة. تتضمن الشبكات العصبية المميزات التالية :

- ضبط التعلم (adaptive learning) : و هو قدرة على تعلم كيفية تأدية المهام حسب البيانات المُعطاة للتدريب أو الخبرة الأولية.
- التنظيم الذاتى (self-organization) : تستطيع ANN أن تنشئ تنظيمها أو تمثيلها الخاص للمعلومات التى تستقبلها خلال زمن التعلم.
- التشغيل فى الوقت الحقيقى (real time operation) : حسابات ANN يمكن تنفيذها على التوازي و بواسطة أجهزة خاصة صُممت وصنعت لتواكب تلك الميزات.
- احتمال العيوب التى تنشأ من تكرار المعلومات (fault tolerance via Redundant information coding) : التحطُّم الجزئى لشبكة

ما يؤدي إلى تناقص الأداء. و مع ذلك تستطيع بعض قدرات الشبكة أن تبقى و تستمر حتى في ظل وجود تحطم كبير في الشبكة.

٩-١-٤ الشبكات العصبية الاصطناعية مقابل البرامج التقليدية

Neural Network Versus Conventional Computers

تتخذ الشبكات العصبية الاصطناعية أسلوباً مختلفاً عن البرامج التقليدية. فالبرامج التقليدية تستخدم طرقاً خوارزمية، بحيث يتبع البرنامج مجموعة من التعليمات لكي يحل مشكلة معينة. و بدون هذه التعليمات لا يستطيع البرنامج أن يحل المشكلة. يحد ذلك من قدرات البرنامج التقليدي، بينما يجب أن تكون البرامج أكثر فائدة و قدرة على فعل ما لا نعرف كيفية فعله.

شبكة الخلية العصبية تعالج المعلومات بطريقة مشابهة لما يفعله عقل الإنسان. فهي تتعلم من الأمثلة التي يجب اختيارها بعناية لكي تعمل الشبكة العصبية بشكل صحيح. تستطيع الشبكة العصبية أن تكتشف كيفية حل المشاكل بنفسها و لذلك من الممكن أن يكون تشغيلها غير متوقع.

على النقيض، تستخدم البرامج التقليدية طريقة لحل المشاكل بحيث لأن يجب معرفة المشكلة و طريقة حلها في تعليمات بسيطة. يتم تحويل تلك التعليمات إلى أحد لغات البرمجة عالية المستوى و منها إلى لغة الآلة التي يفهمها الحاسب.

مع ذلك فإن الشبكات العصبية و البرامج التقليدية يكمل كل منهما الآخر. فهناك مهام تصلح لها البرامج التقليدية و أخرى تناسبها الشبكة العصبية تماماً. و هناك بعض التطبيقات التي يناسبها استخدام الطريقتين معاً، حيث يراقب و يشرف البرنامج التقليدي على الشبكة العصبية لتعمل بكفاءة أكثر.

٢-٩ استخدام الخلايا العصبية الاصطناعية

Using Artificial Neurons

فى هذا الجزء نقترح أكثر من الخلية العصبية الاصطناعية. فنبداً بالتعرف على الخصائص الأساسية للخلية العصبية من خلال مثال مبسط للخلية العصبية. ثم نعرض تشغيل القواعد.

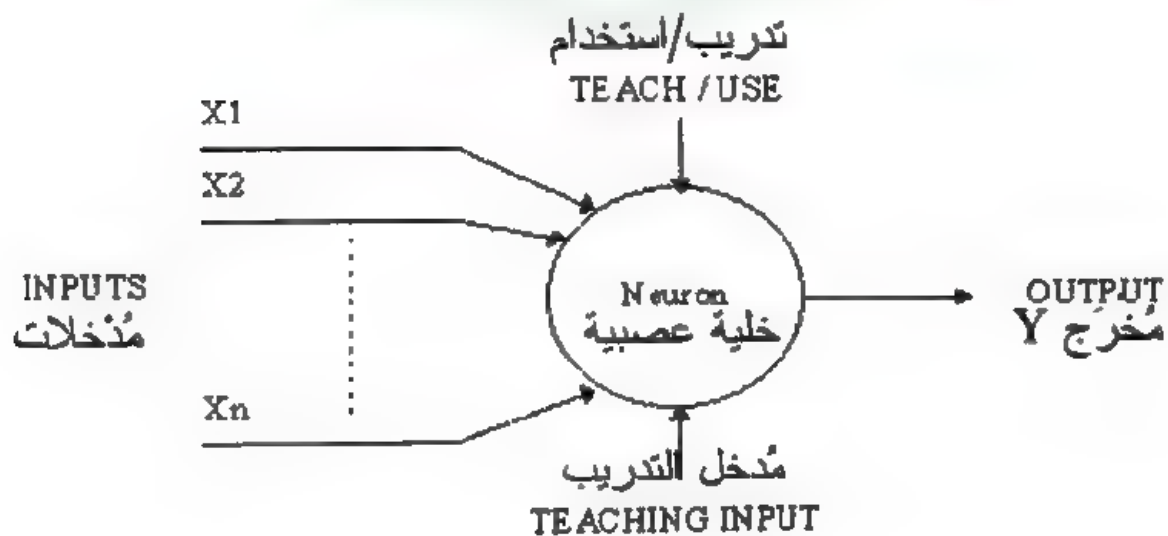
Simple Neuron

١-٢-٩ خلية عصبية بسيطة

الخلية العصبية الاصطناعية هي وحدة (unit) أو عقدة (node) لها عدة مدخلات (X_1, X_2, \dots, X_n) ومُخرج واحد هو Y كما نرى فى شكل ٢-٩. تتلقى الوحدة المدخلات من الوحدات الأخرى (الخلايا العصبية الأخرى) وربما من مصدر خارجي.

تُحسب الوحدة دالة معينة f فى مجموع المدخلات ذوى الأثقال كما يلي :

$$Y = f(X_1 + X_2 + \dots + X_n)$$



شكل ٢-٩ : خلية عصبية بسيطة.

يمكن أن يخدم هذا المخرج كمُدخل إلى الوحدات الأخرى و يكون حاصل الجمع هو المُدخل الصافي (net - input) إلى الوحدة التالية. و يُطَلَق على الدالة F اسم دالة التنشيط أو التفعيل (activation function). و إذا كانت الدالة في أبسط حالتها و هى مجموع المدخلات يُطَلَق على الخلية العصبية اسم الوحدة الخطية (linear unit).

كما رأينا فى شكل ٩-٢، يوجد حالتين لتشغيل الخلية العصبية و هما :

▪ حالة التدريب (training mode)

فى حالة التدريب، يمكن تدريب الخلية العصبية على التنفيذ/أو عدم التنفيذ (fire or not) لنماذج إدخال معينة.

▪ حالة الاستخدام (use mode)

فى حالة الاستخدام، عند اكتشاف نماذج إدخال مثل التى تعلمتها عند مدخل الخلية العصبية، يصبح المخرج المرافق لها هو المخرج الحالى. فإذا كان النموذج المدخل لا ينتمى إلى القائمة التى تدرّبت عليها الخلية، تُستَخدم القاعدة لحساب التنفيذ من عدمه.

Firing Rules

٩-٢-٢ قواعد التنفيذ

قواعد التنفيذ (firing rules) هى أحد المفاهيم الهامة فى الشبكات العصبية و تؤخذ بالحسبان فى تقدير مرونتها العالية. تُقدَّر قاعدة التنفيذ كيف يمكن حساب ما إذا يجب تنفيذ الخلية العصبية على أى نموذج إدخال. و هو يرتبط بكافة نماذج الإدخال و ليس فقط النماذج التى تدرّبت عليها الخلية.

يمكن إعداد قاعدة تنفيذ لخلية عصبية لعمليات منطقية مثل AND و OR

و XOR لمجموعة من المدخلات. نفترض أن لدينا مدخلين فقط هما $X1$ و $X2$

فتكون النتيجة المتوقعة للمخرج Y هي ما نراه في شكل ٩-٣ و هو ما ندرّب عليه الخلية.

AND		
X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR		
X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	1

XOR		
X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0

شكل ٩-٣ : نماذج المخرجات مع AND و OR و XOR.

دعنا الآن نفترض خلية عصبية لها أربعة مدخلات ($X1, X2, X3, X4$) و مخرج واحد Y . و نفترض أن الخلية تم تدريبها على حالات معينة : يكون مخرجها 1 أى يتم تنفيذ القاعدة فى الخلية العصبية (firing rule) عندما تكون قيم المدخلات هي 1111 و 1011 و 1110 و 1010. و يكون مخرجها 0 أى لا يتم تنفيذ القاعدة فى الخلية عندما تكون قيم المدخلات هي 0000 و 0001 و 0010 و 0011. جدول الحقيقة لهذا المثال يكون كما نرى الجدول ٩-١.

جدول ٩-١ : جدول الحقيقة لخلية عصبية معينة.

X1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
X2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
X3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
X4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Y	0	0	0	0	0/1	0/1	0/1	0/1	0/1	0/1	1	1	0/1	0/1	1	1

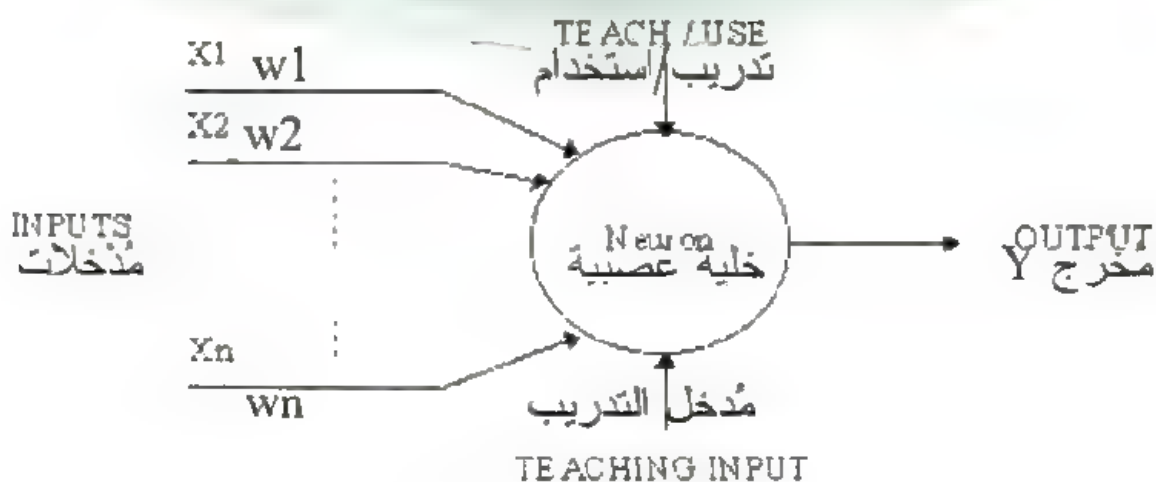
تأمل جدول الحقيقة السابق، لقد تم تدريب الخلية العصبية على ثمانية نماذج إدخال فقط. لذلك عندما يرد نموذج مدخلات إلى خلية عصبية تكون المخرجات فى الحالات الثمانية المتدرب عليها كما نراها فى جدول الحقيقة.

أما الحالات التي لم تتدرَّب عليها الخلية فنجد المخرج 0 أو 1. الآن يجب حساب تطبيق تنفيذ القاعدة أم لا في حالة ورود مدخل جديد. يتم اختيار الحالة الأقرب إلى المدخل الجديد و بنفس الوضع يتم تحديد هل يتم تنفيذ قاعدة الخلية العصبية أم لا. فمثلاً إذا تم إدخال نموذج المدخلات (0111) يكون أقرب نموذج له هو النموذج المعروف (1111) حيث يتم تنفيذ القاعدة و عليه يتم تنفيذ القاعدة على هذا المدخل الجديد و يكون المخرج هو 1 في هذه الحالة أيضاً و يتم تحديث جدول الحقيقة، و هكذا.

٩-٢-٣ خلية عصبية أكثر تقدماً

Advanced Neuron

الخلية العصبية الإصطناعية التي نعرضها هنا لها عدة مُدخلات (X_1, X_2, \dots, X_n) و مُخرج واحد هو Y كما نرى في شكل ٩-٤. تتلقى الوحدة المدخلات من الوحدات الأخرى (الخلايا العصبية الأخرى) و ربما من مصدر خارجي. الجديد هو أن كل مُدخل له ثقل (weight) يمكن تعديله لنمذجة تعلم نقطة الاشتباك العصبي. أثقال مُدخلات الوحدة يمكن أن نرمز لها بالرموز (w_1, w_2, \dots, w_n) .



شكل ٩-٤ : خلية عصبية أكثر تقدماً.

تُحسب الوحدة دالة معينة f في مجموع المدخلات ذوى الأثقال كما يلي :

$$Y = f(w_1 * X_1 + w_2 * X_2 + \dots + w_3 * X_3)$$

إذا زاد المجموع عن قيمة معينة استهلاكية (threshold value) مُحددة من قبل، يتم تنفيذ القاعدة. إضافة النّقل إلى المدخلات يجعل الخلية العصبية أكثر مرونة وقوة. حيث تستطيع هذه الخلية العصبية أن تضبط أدائها بتغيير النّقل و القيمة الاستهلاكية.

٩-٣ عمارة الشبكات العصبية

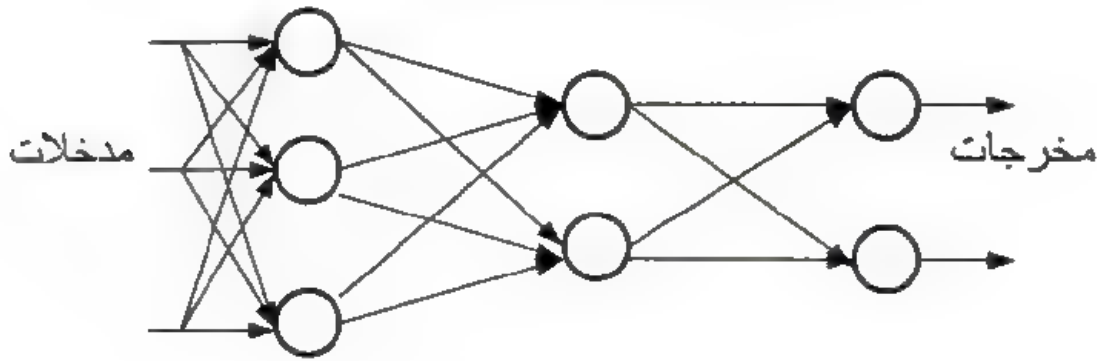
Architecture of Neural Networks

يستخدم الباحثون في مجال الشبكات العصبية أنواع و تقنيات متعددة في تصميم الشبكات العصبية. فيما يلي نعرض لهذه الشبكات.

٩-٣-١ شبكات التغذية الأمامية Feed-Forward Network

شبكات التغذية الأمامية تسمح للإشارات بالمرور في اتجاه واحد فقط (one way) من المدخل إلى المخرج و لا توجد تغذية عكسية (feedback). أى أن المخرج من طبقة معينة لا يؤثر في نفس الطبقة. الشبكات العصبية من هذا النوع تميل إلى أن تكون شبكات ربط المدخلات بالمخرجات مباشرة و تُستخدم على نطاق واسع في التعرف على النموذج. شكل ٩-٥ يعرض مثالاً لهذا النوع.

طبقة الإخراج طبقة مخفية داخلية طبقة الإدخال



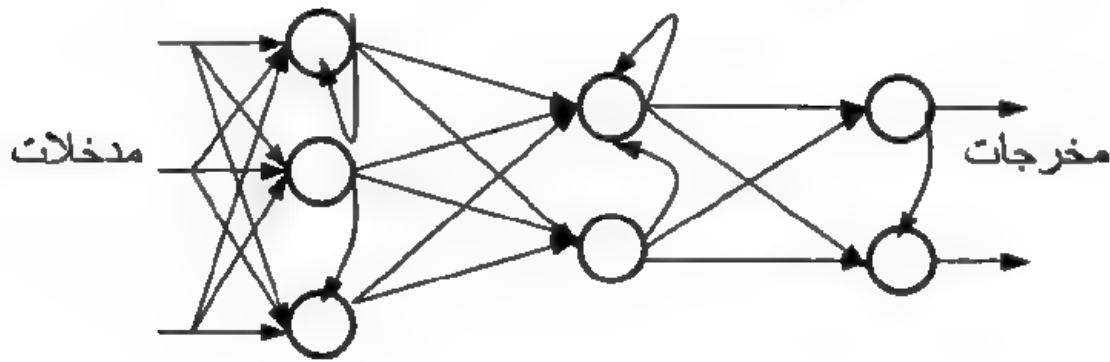
شكل ٩-٥ : مثال لشبكة عصبية ذات تغذية أمامية.

٩-٣-٢ شبكات التغذية العكسية Feed-Forward Network

شبكات التغذية العكسية يمكن أن تحتوي على إشارات تمر في كلا الاتجاهين بإدخال حلقات تكرار في الشبكة. هذا النوع من الشبكات قوى جداً و معقد إلى درجة كبيرة. شكل ٩-٦ يعرض مثلاً لهذا النوع.

شبكات التغذية العكسية متغيرة و ديناميكية، حيث تتغير حالاتها باستمرار إلى أن تصل إلى نقطة التوازن. و تظل عند نقطة التوازن حتى تتغير المدخلات و نحتاج إلى نقطة توازن جديدة. يُشار إلى الشبكات من هذا النوع باسم الشبكات التفاعلية (interactive) أو المرندة (recurrent) على الرغم من أن المصطلح الأخير يُشار به إلى تنظيمات ذات طبقة وحيدة و تغذية عكسية.

طبقة الإخراج طبقة مخفية داخلية طبقة الإدخال



شكل ٩-٥ : مثال لشبكة عصبية ذات تغذية عكسية.

Network Layers

٩-٣-٣ طبقات الشبكة

النوع الأكثر شيوعاً من الشبكات العصبية الاصطناعية يتكون من ثلاث طبقات (layers) أو مجموعات (groups) من الوحدات (units). طبقة وحدات الإدخال (input layer) متصلة بطبقة داخلية مخفية (hidden layer) متصلة بطبقة وحدات الإخراج (output layer) كما نرى في شكل ٩-٤ و ٩-٥.

نشاط طبقة الإدخال يمثل المعلومات الأولية التي تغذي الشبكة العصبية. أما الطبقة المخفية فيُحسَب بأنشطة وحدات الإدخال و تُقَل كل وصلة بين وحدة إدخال ووحدة في الطبقة المخفية. أداء وحدات الإخراج يعتمد على نشاط الوحدات المخفية و تُقَل كل وصلة بين وحدة مخفية و وحدة إخراج.

في التنظيم وحيد الطبقة تكون كل الوحدات متصلة بوحدة أخرى، مُشكِّلةً الحالة الأكثر عمومية و الأكثر قدرة حسابية عن التنظيمات متعددة الطبقات. في التنظيمات متعددة الطبقات، يتم ترقيم الوحدات حسب الطبقة بدلاً من اتباع ترقيم عام.

Learning

٩-٤-١ التعلم

كل شبكة عصبية تمثل المعرفة التي توجد في قيم ثقل الوصلات. تعديل المعرفة المُخزّنة في الشبكة العصبية كدالة في الخبرة يدل ضمناً على قاعدة تعلم لتغيير قيم ثقل الوصلات.

Categorization of Networks

٩-٤-١ تصنيف الشبكات

يتم تخزين المعلومات في مصفوفة الثقل (weight matrix) للشبكة العصبية. عملية التعلم هي عملية حساب الثقل لكل وصلة. طبقاً لطريقة التعلم نستطيع التمييز بين صنفين رئيسيين من الشبكات العصبية هما :

■ شبكات عصبية ثابتة (fixed networks)

في هذا النوع لا يمكن تغيير الثقل في أى وصلة، و تكون قيم الثقل ثابتة و محددة مسبقاً حسب المشكلة المطلوب حلها.

■ شبكات عصبية متكيفة (adaptive network)

في هذا النوع من الشبكات يكون ممكناً تغيير ثقل أى وصلة.

Categorization of Learning

٩-٤-٢ تصنيف التعلم

كذلك يمكن تقسيم تعلم الشبكات العصبية إلى صنفين رئيسيين هما :

■ التعلم تحت إشراف (supervised learning)

يتم دمج مدرب أو مدرس من الخارج بحيث يتم إخبار كل وحدة إخراج عن الاستجابة المطلوبة مع الإشارات الداخلة. من الممكن أن يُطلب معلومات عامة أثناء عملية التعلم. من ضمن هذا النوع من التعلم تعلم تصحيح الأخطاء (error-correction learning).

التعلم دون إشراف (unsupervised learning)

هنا لا يوجد مدرّب أو مدرس من الخارج و يعتمد فقط على المعلومات المحلية. و يُشار إليه بالتنظيم الذاتى، نظراً لأنه ينظم بنفسه البيانات الواردة إلى الشبكة العصبية و يكتشف خصائصها. من هذا النوع طريقة Hebb للتعلم.

٩-٤-٣ سلوك الشبكة العصبية الاصطناعية

Behavior of Artificial Neural Network

سلوك الشبكة العصبية الاصطناعية يعتمد على ثقل كل وصلة ودالة التحويل بين المدخل والمخرج التى تتحدد للوحدات (الخلايا العصبية). يوجد ثلاث أنواع لدالة التحويل التى تحكم العلاقة بين المدخل و المخرج هى :

▪ خطية (linear)

حيث يتناسب نشاط المخرج مع إجمالى المدخل ذو الثقل.

▪ استهلاكية (threshold)

يتم ضبط المخرج عند مستوى واحد أو اثنين (قيمة استهلاكية أولية)، اعتماداً على كون إجمالى المدخل أكبر من أو أقل من القيمة الاستهلاكية.

▪ أسّيّة (sigmoid)

هنا يتغير المخرج بشكل مستمر و لكن ليس خطياً مع تغيرات المدخل. هذا النوع يحاكي الشبكة العصبية البشرية إلى حد ما.

كما درسنا فيما سبق أن الوحدات (الخلايا العصبية) يربط بينها مجموعة من الوصلات. يوجد عدد طبيعى عشرينى مع كل وصلة، هو ثقل الوصلة. نرمز له

بالرمز W_{ij} ليُعبر عن ثقل الوصلة بين الوحدة u_i و u_j . بعد ذلك يتم تحويلها لتمثل نموذج الاتصال في الشبكة بواسطة مصفوفة الثقل W (weight matrix). عناصر مصفوفة الثقل W هي W_{ij} . يوجد نوعين للاتصال : اتصال مثير ذو ثقل موجب و اتصال مانع ذو ثقل سالب. نموذج الاتصال يبين عمارة شبكة الخلايا العصبية.

٩-٤-٤ قاعدة Hebb للتعلّم

Hebb Learning Rule

تنص قاعدة Hebb للتعلّم على أنه يحدث التعلّم من خلال تعديل الاتصالات بين الوحدات أو قيم الأثقال. تشير قاعدة Hebb إلى زيادة الأثقال إذا حدث اتصال بين خليتين فعاليتين في نفس الوقت. و هناك تعديل لتلك القاعدة ينص أيضاً على زيادة الأثقال إذا حدث اتصال بين خليتين غير فعاليتين في نفس الوقت.

قاعدة Hebb تحسّن القدرة الحسابية للشبكة وحيدة الطبقة ذات التغذية الأمامية (feed forward) و التي تُعرف بشبكة Hebb. في هذه الشبكة يوجد اتصال بين وحدة المدخل و وحدة المخرج فقط، لعدم وجود اتصال متبادل بين وحدات المدخلات أو بين وحدات المخرجات.

خوارزم تصميم شبكة عصبية وحيدة الطبقة باستخدام قاعدة Hebb يتطلب عبوراً واحداً خلال مرحلة التدريب و بعد ذلك تتكرر العملية. مع فرض أن المدخل x و المخرج y و المصدر s و الهدف t فيما يلي خطوات الخوارزم :

- Initiate weight with zero values (ضع قيم أولية صفر للأنقال).
 $w_i = 0, i = 1, 2, \dots, n$
- Repeat for each input-output signal (s:t) (كرر لكل إشارة مدخل-مخرج).
 - Activate input units (نشط وحدات المدخل)
 $x_i = s_i, i = 1, 2, \dots, n$
 - Activate output units (نشط وحدات الخرج)
 $y = t$
 - Update weights (عدّل الأتقال)
 $W(\text{new}) = W(\text{old}) + \Delta w$

٩-٥ تطبيقات الشبكات العصبية الاصطناعية

Applications of Artificial Neural Networks

الشبكات العصبية الاصطناعية يمكن تطبيقها افتراضياً في كافة المجالات التي توجد بها علاقة رابطة بين المتغيرات المستقلة (المدخلات) و المتغيرات المتوقعة (المخرجات) و تكون هذه العلاقة الرابطة معقدة. و لقد تم تطبيق الشبكات العصبية الاصطناعية فعلياً في العديد من المجالات منها :

■ اكتشاف الظواهر الطبية (Detection of medical phenomena)

حيث يمكن مراقبة بعض الظواهر الطبية مثل المؤشرات الصحية المختلفة. لقد استخدمت الشبكات العصبية في ملاحظة نموذج التنبؤ المناسب للعلاج لوصفه للمريض.

■ التنبؤ بأسعار الأسهم في سوق المال (Stock Market Prediction)

Prediction : تنبؤ أسعار الأسهم و مؤشرات الأسعار مثال معقد متعدد الأبعاد. استخدمت الشبكات العصبية بواسطة العديد من المحللين الفنيين لقياس تنبؤات عن أسعار الأسهم مقابل عدد من العناصر المؤثرة مثل الأداء السابق للسوق و المؤشرات الاقتصادية.

▪ **منح القروض الائتمانية (Credit Assignment) :** أجزاء متنوعة من المعلومات تُعرَف عادةً عن العميل المتقدم للحصول على قرض ائتماني. من هذه المعلومات عمر العميل و تعليمه و عمله و عناصر أخرى ممكن أن تكون متاحة. بعد تدريب الشبكة العصبية على البيانات التاريخية، تستطيع الشبكة العصبية أن تحلل أغلب مواصفات العميل و تستخدمها لتصنيف العميل كعميل جيد أو عميل سيئ.

▪ **مراقبة ظروف تشغيل الآلات (Monitoring the Condition of Machinery) :** ممكن للشبكات العصبية أن تخفض الأسعار باستغلال الخبرة في جدولة الصيانة الوقائية للآلات. حيث يمكن تدريب الشبكة العصبية للتمييز بين أصوات الماكينات عندما تعمل بشكل طبيعي و عندما يكون هناك مشكلة. بعد فترة التدريب تلك، يمكن استخدام خبرة الشبكة لتحذير الفنيين عن العطل القادم مستقبلاً قبل حدوثه و تسببه في عطل ضخم أو كارثي.

▪ **إدارة المحركات (Engine Management) :** نستخدم الشبكات العصبية هنا في تحليل المدخل إلى حساسات القياس (sensors) من محرك ما. تتحكم الشبكة العصبية في العناصر المؤثرة المختلفة حيث يعمل المحرك، لكي تتجز هدفاً ما مثل خفض استهلاك الوقود.

٦-٩ أسئلة

١. ما هي مميزات الجهاز العصبي الطبيعي؟
٢. وضّح بإيجاز مفهوم شبكات الخلايا الاصطناعية.

٣. ما هي مميزات شبكات الخلايا الاصطناعية؟
٤. قارن بين الشبكات العصبية الاصطناعية و البرامج التقليدية؟
٥. وضّح مستعيناً بالرسم تركيب و عمل الخلية العصبية neuron، مبيناً الحالات التي يمكن أن تمر بها الخلية في تشغيلها.
٦. وضّح مستعيناً بالرسم عمارات الشبكات العصبية.
٧. ما هي أنواع الشبكات العصبية طبقاً لطريقة التعلم؟
٨. ما هي أنواع تعلم الشبكات العصبية؟
٩. اشرح أنواع دوال التحويل في الشبكات العصبية؟
١٠. اشرح بإيجاز بعض تطبيقات الشبكات العصبية.



الفصل العاشر

المعينات الإلكترونية و التعليم

Agents & Teaching

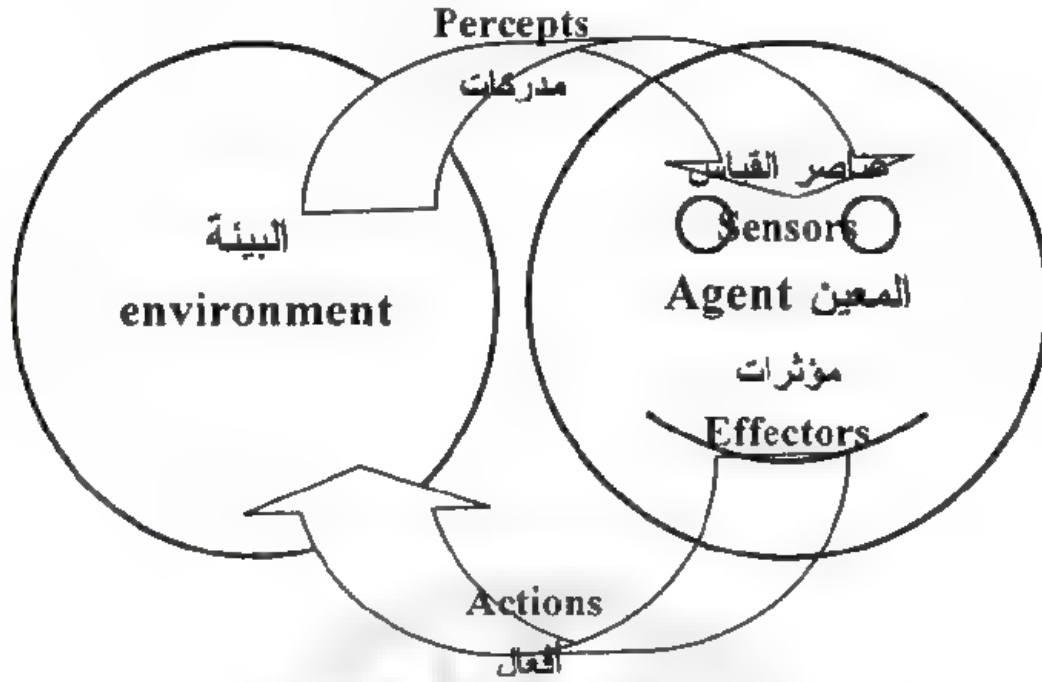
من أحدث الفروع الناشئة في مجال الذكاء الاصطناعي هو المعينات الإلكترونية البرمجية (Software Agents). و هي برامج تؤدي وظيفة (أو وظائف) معينة بناءً على وقوع حدث أو أحداث معينة. و تُعدّ هذه الوظائف (الإجراءات) بمثابة ردود أفعال لتلك الأحداث. و تخدم المعينات في العديد من البرامج و بشكل كبير في النواحي التعليمية منها.

Introduction

١-١٠ مقدمة

المعِين (Agent) هو أى شيء يمكن أن يُلاحظ على أنه يدرك البيئة الموجود بها من خلال عناصر قياس (sensors) لقياس الأشياء المُدرَكَة التي تُسمّى المُدْرَكَات (percepts) و من ثمّ اتخاذ أفعال أو إجراءات (actions) باستخدام المؤثرات (effectors).

المعِين البرامجي (Software Agent) هو نفس المعِين السابق مع برمجة عناصر القياس والمؤثرات. شكل ١-١٠ يبين تعامل المعِين مع البيئة التي يعمل عليها.



شكل ١٠-١ : تعامل المعين مع البيئة التي يعمل خلالها.

١٠-٢ طبيعة المعين الذكي

The Nature of Intelligent Agent

المعين الذكي (Intelligent Agent) هو معين برامجي لابد أن ينجح دائماً في أداء عمله بكفاءة و سرعة. الغرض من برامج المعين هو بناء معين ذكي له مواصفات متشابهة مع غيره من المعينات حيث تقبل المدركات أو الاستشعارات من البيئة لتقوم بإجراء الفعل المناسب. جدول ١٠-١ يعرض بعض نماذج للمعينات.

Performance Measure

١٠-٢-١ معيار الكفاءة للمعين

معيار/مقياس الكفاءة للمعين الذكي : لا يوجد معيار ثابت لأي معين. ولكن المعيار هو مدى نجاح المعين في أداء عمله. و توجد أربع عناصر تحدد مدى معقولية المعين الذكي و هي :

- مقياس الأداء الذي يحدد درجة النجاح.
- ماذا يستطيع أن يدرك من البيئة المحيطة؟
- ماذا يعرف عن البيئة؟
- ما هي الأفعال التي يمكنه أدائها؟

سوف يُقاس الأداء حسب النتائج. و لا يمكننا إلقاء اللوم على معين لفشله في إدراك شيئاً ما أو اتخاذ فعل معين غير قادر على أدائه. لكن المعين الذكي يجب أن يفعل الأشياء الصحيحة دائماً.

جدول ١٠-١ : نماذج لبعض المعينات الإلكترونية

نوع المعين	المدرجات Percepto	الأفعال Actions	الهدف	البيئة
معين تصنيف أمراض النبات	الأعراض	فحص أجزاء النبات وبعض الاختبارات	علاج النباتات و إنتاج عالي الجودة بتكلفة أقل ما يمكن	المزرعة و المزارعين
معين تصنيف صور	النقطة الكثافة اللون	طباعة تصنيف الصورة	ضبط تصحيح التصنيف	صور من مجال التطبيق
معين معلم لغة برمجة	الأوامر و الجمل	طباعة التصحيحات والاقتراحات	تحسين مستوى الطالب	الطلبة
معين تصحيح تحرير	الفقرات و الجمل و الكلمات و الهوامش و المسافات و النقاط	تحويل أول حرف و ضبط الهوامش و تصحيح الكلمات	زيادة الدقة والأمان	برنامج تحرير النصوص

Agent Design

١٠-٢-٢ تصميم المعين

يمكن تمثيل المعين بدالة واحدة بسيطة لها مدخلات و مخرجات. المدخلات تمثل المُدْرَكَات و المخرجات تمثل الأفعال. شكل ١٠-٢ يبين نموذجاً لدالة عادية تقوم بعملية حسابية معينة و تنتج لنا مُخْرَجاً إذا تَوَفَّر لها مُدْخَلٌ.

المُدرّك هو العدد: //X Function Cubic (X)

$Z = X * X * X$

الفعل هو مكعب العدد المُدرّك X و هو : // Z Return Z

End

شكل ١٠-٢ : مثال لدالة تماثل عمل المعين.

أداء المعين هو الفعل الذى يؤدّى بعد إدراك تسلسل من المدركات فى بيئة عمل المعين. وظيفة AI هى تصميم برنامج المعين (Agent Program) أى الدالة التى تقوم بعملية التحويل من الإدراك إلى الأفعال و هذا البرنامج يتم تشغيله على جهاز حاسب آلى (عمارة الحاسب Computer Architecture). هذه العمارة إما أن تكون حاسب آلى عادى (عام الغرض) أو حاسب آلى خاص الغرض و عليه برمجيات تربط/و تفصل بين البرنامج و العمارة. لذا يمكننا صياغة المعادلة التالية:

معين = برنامج + عمارة حاسب || Agent = Program + Architecture

كذلك يلزم عند تصميم برامج المعين الذكى معرفة ما يلى :

- ما هى الأشياء التى يدركها البرنامج؟ Percepts.
- ما هى الأفعال التى تتم بناءً على ذلك؟ Actions.
- ما هى الأهداف (أو مقياس الأداء) التى يُفترض أن يقوم بها المعين؟ Goals.

Agent Environments

١٠-٢-٣ بيئات المعين

يجرى تصميم المعين لي عمل من خلال بيئة معينة يتأثر بها و يؤثر فيها. لذلك فإن نوع البيئة يؤثر في تصميم المعين. من الممكن أن نحصر أنواع البيئات التي يُتَوَقَّع أن يعمل عليها المعين فيما يلي :

• بيئة قابلة للتناول/غير قابلة للتناول (Accessible/Inaccessible) :

إذا استطاع المعين أن يجمع الحالة الكاملة للبيئة فإن البيئة تكون قابلة للتناول أو تكون البيئة غير قابلة للتناول.

• بيئة قابلة للحل/غير قابلة للحل (Deterministic/

NonDeterministic) : إذا أمكن حساب الحالة التالية من الحالة الحالية

تكون البيئة قابلة للحل أو تكون البيئة غير قابلة للحل.

• بيئة قابلة للتسلسل/غير قابلة للتسلسل (Episodic/Non-Episodic) :

إذا أمكن تقسيم المعين إلى سلسلة من الأحداث تكون البيئة قابلة للتسلسل أو

تكون البيئة غير قابلة للتسلسل.

• بيئة ثابتة/متغيرة (Static/dynamic) : إذا أمكن تغيير البيئة أثناء

عمل المعين تكون البيئة متغيرة أو تكون بيئة ثابتة.

• متصلة/غير متصلة (تمثيلي/رقمي) (Continuous/Discrete) : إذا

كان هناك عدد محدد من المدركات تكون البيئة غير متصلة أو تكون البيئة متصلة.

• ملاحظة كاملة/جزئية (Fully/Partially Observable) : إذا استطاع

المعين أن يدرك جميع عناصر حالة البيئة التي يعمل عليها تكون البيئة ملاحظة كاملة أو تكون ملاحظة جزئياً.

• معين واحد/مجموعة معينات (Single/Multi Agents) : في حالة

وجود أكثر من معين يعملون بشكل جماعي و يتبادلون المعلومات تكون البيئة متعددة المعينات أو تكون البيئة ذات معين واحد.

١٠-٢-٤ خصائص المعين الذكي

Intelligent Agent Attributes

توجد ستة خصائص رئيسية للمعينات الذكية تُميّزها عن الأنواع الأخرى لتطبيقات البرامج و هي :

• الاستقلالية (Autonomy)

المعين الذكي يجب أن يكون لديه المقدرة على القيام بالأفعال المؤدية إلى اكتمال هدف أو مهمة (أهداف أو مهام) معينة دون دفع أو حفز أو تدخل المستخدم النهائي. و يجب أن يكون عنصر مستقل و لديه تحكم في الحالة الداخلية و السلوك.

- **براعة الاتصال (Communication Ability)**

المعين الذكى يجب أن يتناول المعلومات من مصادر أو معينات أخرى عن الحالة الحالية للبيئة التى يعمل خلالها. من الممكن أن يكون هذا الاتصال فى شكل طلب مفرد مع مجموعة بسيطة مختصرة من الردود أو اتصال مركب مع ردود متغيرة.

- **المقدرة على المشاركة (Capacity for Cooperation)**

المعين الذكى يجب أن يكون لديه روح المشاركة ليبقى و ينجح. أى أنه يجب أن تعمل المعينات الذكية بشكل جماعى لإنجاز المهام المعقدة أو الصعبة.

- **المقدرة على الوصول إلى الحل (Capacity for Reasoning)**

أحد سمات المعين الذكى هى القدرة على التفكير و الاستنتاج و الاستدلال على الحل. يتم ذلك اعتماداً على قواعد أو معرفة معينة.

- **ضبط الأداء (Adaptive Behavior)**

المعين الذكى يجب أن يكون قادراً على فحص البيئة التى يعمل عليها و نجاح الأفعال السابقة التى تمت بناءً على مُذكرات مماثلة مع القدرة على ضبط و تهيئة هذه الأفعال لزيادة احتمالية النجاح فى تحقيق أهدافهم.

- **جديرة بالثقة و يُعتمد عليها (Trustworthiness)**

المعين الذكى يجب أن يكون جديراً بالثقة فى دقته و قدرته على تحقيق الهدف المرجو منه.

١٠-٢-٥ المعين مُعَلِّم و مُتَعَلِّم

Teaching and Learning Agent

المعينات المُعَلِّمَة (Teaching Agents) تُسْتَخْدَم في تدريس الموضوعات المختلفة باستخدام أساليب مختلفة. أحد الأساليب المشهورة هو تقنية المُدَرِّب (Coach). حيث يقوم المُدَرِّب (المعين) بمراقبة أفعال الدارسين المُتَدَرِّبين ثم يقدِّم له تعليقاً أو نصيحة لتصحيح أخطائهم. أحد مميزات المعين المُدَرِّب هو أن التدريب يُصَبِّح فائدته أكبر من تكلفته نظراً لأن المُدَرِّب متاحاً حسب رغبة و على راحة المُتَدَرِّب/المُتَدَرِّب، فيوفر ذلك من وقت المُتَدَرِّب/المُتَدَرِّب. من هنا يتعلم الدارس من أخطائه و يجد الوسيلة ليتخلص منها مستقبلاً.

المعينات المُتَعَلِّمَة (Learning Agents) هي معينات قادرة على ربط (خياطة) أدائها بخيارات الأفراد عن طريق التعلم من الأداء السابق للمستخدمين. تأثير التعلم مطلوب للدقة المتزايدة للتوقعات. يتطلب ذلك إدخال معلومات أكثر من المستخدم. بعض المعينات تتضمن قدرتها على التعلم على تطبيقات الحاسب التي تجعل أداء المهام المكررة يتم الياً. حيث تعلَّم المعين عن طريق مراقبة أفعال المستخدم و تحديث المعلومات عن المستخدم و مستواه في التعلم.

١٠-٣ تطبيقات المعين الذكي

Intelligent Agent Applications

تتنوع تطبيقات المعين الذكي في مجالات عديدة يمكننا حصرها في تصنيفات رئيسية. كل صنف يضم مجموعة من التطبيقات. فيما يلي نتعرَّف على بعض التصنيفات مع إمكانية جمع بعض التصنيفات في معين واحد.

Agents for Education

١٠-٣-١ معينات تعليمية

تقوم هذه المعينات بمساعدة المستخدمين في التعلم و التوجيه و منها :

- **معينات مُدرّبة (Tutors Agents)**

هذه المعينات كمساعدة للدارسين على تعلّم الخبرات و العثور على المعلومات و التعرف على نقاط الضعف.

- **معينات تخطيط مهنية و بحث عن الوظائف (Career Planning and Job Search Agents)**

تساعد هذه المعينات المهنيين المتخصصين على إدارة أهداف مهنتهم وتخصصهم.

- **معينات تدريبات معملية (Laboratory Exercises Agents)**

هذه المعينات ترشد الطلبة في التدريبات المعملية و توجههم على كيفية الانتقال من مرحلة لأخرى.

١-٣-٢ معينات واجهة اتصال ذكية Smart Interface Agents

يُعتبر مجال واجهة اتصال المستخدم بالحاسب الآلى مجالاً خصباً لتطبيقات المعين الذكى. استخدام تعليمات التشغيل (wizard) كأيقونات أو روابط أو قوائم فى معظم منتجات MS Office مثلاً على ذلك. و تتضمن تطبيقات المعينات الذكية فى هذا المجال ما يلى :

- **واجهات اتصال مُدرّبة (Interface Tutors)**

هذه المعينات توفر تحسناً للاتصال بالنظم عن طريق ملاحظة كيفية أداء المستخدم و تصحيح أخطائهم.

- **مساعدات جدولة (Scheduling assistants)**

هذه المعينات تساعد على توفير الوقت و إدارة النشاط المشترك للمستخدمين.

• **معينات بحث (Search Agents)**

هذه المعينات تساعد على العثور على ملفٍ ما أو استخراج معلومات جديدة من مصادر غير المستخدم.

• **معينات عرض أو تقارير (Presentation and Report Agents)**

هذه المعينات تُستخدَم في عرض المعلومات للمستخدم في التصميم الذي يفضلُه المستخدم.

• **معينات تصفُّح (Navigation Agents)**

هذه المعينات توفر الطرق التي يفضلها المستخدم لعرض المعلومات و تصفُّحها من خلال مسارات إلى تلك المعلومات.

Development Agents ١٠-٣-٣ معينات تطوير

تقوم هذه المعينات بمساعدة المستخدمين في تطوير و بناء الحلول الملائمة لمشاكل التطوير أو التصميم.

• **معينات تحليل و تصميم (Analysis and Design Agents)**

هذه المعينات يستخدمها محللو النظم لفهم طبيعة احتياجات التطبيقات و بناء التصميم المناسب.

• **معينات اختبار (Testing Agents)**

هذه المعينات يمكن تطبيقها خلال تطوير التطبيق بغرض الاختبار و الفحص.

• معينات تجهيز حزم (Packaging Agents)

هذه المعينات لتحزيم (لتطويق) التطبيقات بالتسهيلات الملائمة للبيئات التي تعمل وقت التشغيل.

• معينات مساعدة (Help Agents)

هذه المعينات تهدف إلى تقديم المساعدة للمستخدم في حالة كون المساعدة متغيرة بشكل كبير جداً و صعبة البناء.

Agent Structure

١٠-٤ تركيب المعين

يتدرج و يتنوع تركيب برنامج المعين من دالة بسيطة إلى نظام كامل. لكن الأساس واحد في أن المعين لديه عناصر قياس (Sensors) يستخدمها في إدراك المدركات (Percepts) و لديه عناصر المؤثرات (Effectors) التي يستخدمها لأداء أفعال و إجراءات (Actions) مناسبة.

١٠-٤-١ دالة النموذج العام للمعين

الأساس في المعينات قدمه Russell و Norvig (٢٠٠٣) و يمكن أن نراه في دالة النموذج العام الذي يُطلق عليه الإطار (Frame) الموجودة في شكل ١٠-٣. نرى في الشكل ١٠-٣ اسم المعين هو اسم الدالة Frame_Agent تستقبل الشيء المُدرك percept و ترجع بالفعل action. السطر الثاني بالدالة يُعرّف المتغير memory و هو الذاكرة التي تحتوى على معلومات المعين عن العالم. أما السطور من الثالث إلى الخامس فتستدعى دوال في مجموعها تقوم بعمل المعين.

السطر الثالث يستدعى الدالة UPDATE-MEMORY التي تأخذ معلومات المعين عن العالم (memory) و معه الشيء المُدرك (percept) و تقوم بتعديل الذاكرة memory بإضافة percept إليها و ترجع بالذاكرة المُعدّلة

memory المضاف إليها المدرك الجديد. السطر الرابع يستدعي الدالة CHOOSE-BEST-ACTION التى تختار أو تحسب أحسن فعل action مناسب و ترجع به فى المتغير action. السطر الخامس يستدعي الدالة UPDATE-MEMORY مرة أخرى التى تأخذ معلومات المعين عن العام (memory) و معه الفعل الجديد (action) المناسب للمدرك الجديد (percept) و تضيفه إلى الذاكرة memory و ترجع بالذاكرة الجديدة memory أيضاً. آخر سطر يرجع بالفعل أو الإجراء الذى أوجدته دالة المعين و هو action.

```

Function Frame_Agent (percept) returns action
static : memory, the agent's memory of the world
memory ← UPDATE-MEMORY(memory,percept)
action ← CHOOSE-BEST-ACTION(memory)
memory ← UPDATE-MEMORY(memory,action)
return action
    
```

شكل ١٠-٣ : دالة هيكل المعين.

كما نرى فى هذا النموذج العام يحدث ما يلى :

- تعديل ذاكرة المعين لتعكس المدرك أو الاستشعار الجديد.
- اختيار أحسن فعل أو إجراء مع تحديد الحقيقة أو الحقائق التى تم بناء هذا الاختيار عليها.
- تعديل ذاكرة المعين بإضافة الإجراء وسببه.

كذلك يوجد بعض القصور فى هذا المعين و هو :

- يستقبل مدرك واحد فقط وليس سلسلة من المدركات.
- لا يحتوى على الهدف "مقياس الأداء" و الحكم يكون خارجياً على الأداء.

١٠-٤-٢ معين جدول المتابعة

كذلك قدم Russell و Norvig (٢٠٠٣) معين جدول المتابعة، الذي تقوم فكرته على إنشاء وتعبئة جدول عن كافة المدركات والإجراءات المناسب لكل مدرك أو مجموعة من المدركات. بحيث يكون كل مُدْخَل في الجدول يحتوى على مدرك أو مجموعة من المُدْرَكَات يقابلها فعل أو مجموعة من الأفعال. و عندما يدرك المعين قيمة في خانة المدركات في مُدْخَل معين يستخرج المعين القيمة المقابلة لها في خانة الأفعال في نفس المُدْخَل. شكل ١٠-٤ يعرض نموذجاً لهذا المعين.

```

Function Table_Agent (percept) returns action
static : percepts, a sequence initially empty
          table, a table indexed by percept sequences,
              initially fully specified
do
    append percept to the end of percepts
until no new percepts
action ← LOOKTABLE(percepts, table)
return action
    
```

شكل ١٠-٤ : دالة معين جدول المتابعة.

نرى في الشكل ١٠-٤ اسم المعين هو اسم الدالة Table_Agent تستقبل الشيء المُدْرَك percept و ترجع بالفعل action. السطر الثاني بالدالة يُعرّف المتغيرين percepts (و هو جدول فارغ لتسجيل تسلسل المدركات) و table (و هو جدول مفهرس طبقاً لتسلسل المدركات مع الأفعال) أما السطرين الثالث و الرابع فيمثلان عمل المعين.

السطر الثالث يلحق أى مدرك جديد بجدول تسلسل المدركات percepts الذى يكون فارغاً فى البداية. يتم ذلك من خلال حلقة تكرار تملأ الجدول

percepts بالمدركات. أما السطر الرابع فيستدعي الدالة LOOKTABLE التي تختار الفعل action المقابل من جدول table وترجع به في المتغير action.

ينجح هذا المعين مع التطبيقات التي تحتوى على عدد محدود من المدركات و حسب تسلسل المدركات يتم انتقاء أحسن فعل. و لكنه لا يصلح مع التطبيقات التي تحتوى على عدد لا نهائى من المدركات لأنه فى هذه الحالة نحتاج إلى جدول ضخم جداً يتطلب ما يلى :

- يحتاج إلى حيز تخزين وذاكرة كبيرة.
- البحث فى الجدول يستغرق وقتاً طويلاً.
- يحتاج إلى وقت كبير من المصمم.
- المعين لا يفعل شئ سوى اختيار الفعل المقابل.
- حتى وإن تعلم المعين فإن ذلك يستغرق وقت طويلاً.

١٠-٤-٣ معين يدرك ما حوله من العالم

قدّم Russell و Norvig (٢٠٠٣) هذا المعين ليدرك و يلم بحالة العالم حوله ثم يقوم بإضافة المُدرك الجديد إلى الحالة السابقة منتجاً حالة جديدة هي حالة العالم الحالية ويحول هذه الحالة إلى متغير حالة/شرط ثم يبدأ فى البحث عن قاعدة ما فى مجموعة من القواعد الشرطية (قواعد الشرط-الفعل) حتى يجد قاعدة مطابقة فى الشرط فيستخرج الفعل المقابل للشرط ويرجع به. شكل ١٠-٥ يوضح دالة هذا المعين.

نرى فى الشكل ١٠-٥ اسم المعين هو اسم الدالة Recognize-World التي تستقبل الشئ المُدرك percept و ترجع بالفعل action. السطر الثانى بالدالة

يُعرّف المتغيرين state (و هو وصف لحالة العالم الحالية) و rules (و هي زمرة قواعد الشرط-الفعل). أما السطور من الثالث إلى السادس فيمثلان عمل المعين.

Function Recognize-World (percept) returns action
static : state, a description of the current world state
 rules, a set of condition-action rules

state ← UPDATE-STATE(state, percept)
 rule ← RULE-MATCH(state, rules)
 action ← RULE-ACTION(rule)
 state ← UPDATE-STATE(state, action)
 return action

شكل ١٠-٥ : دالة معين يدرك ما حوله من العالم.

السطر الثالث يستخدم المدرك الجديد في تحديث معلومات المعين و ذلك بإضافته إلى الحالة القديمة state لينتج وصف الحالة الجديدة الحالية state أيضاً و ذلك عن طريق استدعاء الدالة UPDATE-STATE. أما السطر الرابع فيستدعي الدالة RULE-MATCH التى تبحث في قاعدة الشرط-الفعل rules عن حالة قريبة من نفس الشرط state و ترجع بتلك الحالة أو القاعدة rule. السطر الخامس يستدعي الدالة RULE-ACTION التى تستخرج الفعل الموجود action فى القاعدة rule و ترجع به. و يستدعي السطر السادس الدالة UPDATE-STATE مرة أخرى ليسجل القاعدة الجديدة (state, rule) فى زمرة القواعد rules و ترجع بالحالة الأحدث التى تصبح الحالة الحالية state. السطر الأخير يرجع بالفعل المناسب.

١٠-٥ الاتجاهات و التطويرات فى تقنية المعينات

نَعرِف هنا على الاتجاهات و التطويرات فى مجال المعينات الإلكترونية. حيث توجد عدة عوامل ترتبط بتقنية المعينات و تطويرها حالياً و مستقبلاً. العامل الأول هو الروابط بين التطورات فى مجال الحاسب و تقنية المعينات. العامل الثانى هو العامل البشرى (المستخدم) فى تطوير المعينات. العامل الثالث هو الموردون و المطورون للمعينات. العامل الرابع هو الدولة أى الحكومة و ما يمكن أن تحصل عليه الحكومة من مكاسب من خلال استخدام تقنية المعينات. العامل الأخير هو التطورات المتلاحقة فى الإنترنت و حولها. نبدأ أولاً فى دراسة هذه العوامل فى الوقت الحالى و ما سبقه و فيما يلى نلقى نظرة على تلك العوامل و علاقتها بتقنية المعينات.

• الحاسب و تقنية المعين (Computer and Agent Technique)

التطورات المتلاحقة فى الإنترنت و ما حولها مشابهة تماماً لتطورات الحاسب و واجهات الاتصال بهم. زاد انتشار الحاسب لظهور وسيلة الاتصال البيانية (GUI) بالقوائم و الرسوم بدلاً من الأوامر القديمة. تستخدم أغلب GUIs و معها تطبيقات الإنترنت تعليمات التشغيل wizard و هى أساساً عمليات صغيرة تتم فى خلفية أعمال أخرى متزامنة. هذه التعليمات تعتبر البداية البسيطة للمعينات الحقيقية و هى موجّهة عادة بواسطة مجموعة من قواعد (if-then) و هى ليست ذكية جداً و ليست مستقلة.

• المستخدم (The User)

حتى الآن أغلب مستخدمي المعينات هم من الباحثون و جزء من مستخدمي شبكة WEB. لكن العامل الرئيسى فى نجاح المعينات هو قبول المستخدم

للمعين و من ثمّ طلبه. تستطيع المعينات جعل استخدام الإنترنت و الحاسب بصفة عامة أكثر سهولة وقابلية لدى المستخدم. حتى هذا الوقت خرج العديد من الأبحاث فى مجال المعينات الذكية. هذه المعينات الغرض الرئيسى منها هو توفير واجهة اتصال سهلة الاستخدام مع النظم المعقدة. باستخدام تلك الأشياء كعلامات متحركة تصبح جميع النظم ذات مظهر جذاب و سهل للمستخدم العادى و الخبير على حد سواء.

• الموردون و المطورون (Suppliers and Developer)

ظهور الإنترنت ضاعفت الطلب على برامج المعينات الذكية. من الناحية الوظيفية، نقل استخدام الإنترنت من مجرد التصفح إلى وسط نقل المعلومات من نقطة إلى أخرى بكفاءة. هذا الاتجاه وجّه تطوير المعينات الذكية من بيانات البحث الأكاديمى إلى الاستخدام التجارى الواسع. علاوة على ذلك يلعب موردو المعلومات و/أو الخدمات دوراً مزدوجاً بعدما أصبحوا منتجياً أيضاً و هو ماله التأثير على التطويرات فى تقنية المعينات.

السمات التى كانت أهميتها ثانوية مثل ربحية التقنية و ما إذا كانت تلبي طلبات المستخدم أو سوق معين أصبحت الآن قضايا جوهرية. القائد التجارى يريد أن تختلف خصائص المعينات بالمقارنة مع الأبحاث الأكاديمية. و فى الحقيقة لا يجب أن تظل المعينات إلى الأبد مجرد كائنات شبيهة بحثياً و كذلك لا يمكن أن يُهَدَف من المعينات أن تحقق نتائج سريعة جداً.

و لقد وجد المطورون أن تطوير و دعم تكاليف المعينات مثل غيرها من التطبيقات. يتوقع المطورون زيادة تكلفة المعينات بمجرد أن

تصبح نقالة (mobile) بصرف النظر عن استخدام نموذج معين واحد أو عدة نماذج. فى بعض الأسواق مثل إدارة الشبكات نجد أن المعينات من المنتجات المطلوب شرائها.

• الحكومة (The Government)

الدخل المنخفض ليس المانع لبعض المجتمعات من استخدام قناة معلومات رئيسية لكنه نقص التعليم و المهارات. يمكن استخدام المعينات لملا هذه الفجوة و لعدم حصر اهتمام الحكومة على توفير احتياجات جزء صغير من المواطنين من مجتمع المعلوماتية. وهناك حقيقة هامة أيضاً و هى أن الحكومات هى أحد الموردين الكبار للمعلومات و أحد المستهلكين الكبار لها أيضاً. فهى التى تتفق على المشروعات الكبيرة و لديها الأمل فى أن يأتى ذلك بالتقنيات و التطبيقات الهامة.

• الإنترنت (The Internet)

عدد مستخدمى الإنترنت ينمو بسرعة كبيرة و كذلك عدد من يقدمون الخدمات و المعلومات عبر الإنترنت و هى أحد التطويرات الجديدة بالملاحظة. و هذا يشير إلى ازدياد حاجة المستخدمين للتقنيات المختلفة (أو بالأحرى للخدمات) لإنجاز مهام معينة على الإنترنت.

و لقد قفز مطورو برمجيات التصفح فى هذا الاتجاه بإنشاء حزم برامج متعددة الفوائد و الاستخدامات. تستطيع المعينات تقديم تلك الوظيفية بشكل أفضل فتوفر استقلالية رائعة بين البرامج و الأجهزة و توسع فى الوظيفية و المرونة العالية. لذلك فإن شبكة Web هى الخطوة الأولى الجبارة نحو نظم أكثر تطوراً و تعقيداً (مثل المعينات الذكية) و نحو إنشاء مواصفات قياسية مفتوحة للإنترنت.

و مع ذلك فإن تقنية المعينات التى تسببت فى ذلك سوف تتغير لتساعد فى مهام جديدة مثل : تنفيذ المهام مثل البحث (محلياً) على قدر الإمكان و إرسال نتيجة البحث للمستخدم عبر الإنترنت. أيضاً، استخدام نتائج و تجارب المهام المنفذة سابقاً لعمل تنفيذات مستقبلية لنفس المهمة و لكن بكفاءة أكبر. و أخيراً استخدام ذكاء المعينات لإنجاز مهام خارج نطاق ساعات الذروة و توزيع العبء.

١٠-٦ المعين المرشد للمتدربين على لغة C++

An Advisor Agent for C++ Trainees

فى هذا الجزء نقدم أحد التطبيقات للمعينات الإلكترونية (ناصر، ٢٠٠٤) و هو عبارة عن معين ذكى مرشد للمتدربين على لغة C++ (An Intelligent Advisor Agent for C++ Trainee).

Advisor Agent

١٠-٦-١ المعين المرشد

نعلم أن المعين هو شىء ما يلاحظ/يدرك و يعمل على بيئته. أما المعين الذكى فهو برنامج يحدد الإجراءات و الأفعال اللازم القيام بها عند إدراك تسلسل من الأفعال و هو عبارة عن مكونات برمجية تتحد معاً ليكون لديها القدرة على العمل بذكاء. من الممكن أن تحتوى المعينات الذكية على معرفة أكثر عن الاحتياجات و الأفضليات و نموذج أداء الشخص أو العملية.

المعينات الذكية يجب أن يكون لديها القدرة على ضبط الاستدلال على الحلول. كذلك يجب أن تكون قادرة على تناول المعلومات من مصادر أو معينات أخرى و إنجاز أفعال تؤدي إلى اكتمال مهمة معينة. و يجب أن تكون قادرة على تفحص البيئة الخارجية و مدى نجاح الأفعال التى تم اتخاذها فيما سبق فى ظروف مشابهة و ضبط تلك الأفعال.

نظام التعليم الذكى هو برنامج على الحاسب الآلى لدعم العملية التعليمية التى يمكن أن تشخص مشاكل التدريب الشخصية. هذه القدرة على التشخيص تمكنه من ضبط التعليمات و التوجيهات حسب حاجة المُتَدَرِّب/ة/ المُتَدَرِّب. تحتاج المُتَدَرِّب/ة/ المُتَدَرِّب إلى المساعدة فى اكتساب و فهم المعرفة المعروضة. لذا فقد تحول دور الحاسب من مجرد ناقل للمعرفة إلى أداة تساعد فى بناء المعرفة.

لقد أصبحت لغة C++ من اللغات الواسعة الانتشار و خصوصاً فى برامج التقنيات المُوجَّهة الأهداف (Object-Oriented Techniques) و برامج الذكاء الاصطناعى. لذلك كان من المهم بناء معين مرشد للمُتَدَرِّبين على هذه اللغة.

يعمل برنامج المعين الذى نعرضه على مساعدة و إرشاد المُتَدَرِّبين على لغة C++. فهو يتضمن معرفة مبدئية أولية بالإضافة إلى معرفة عن خبرة و أداء المُتَدَرِّب/ة/ المُتَدَرِّب. و هو قادر على ضبط قاعدة المعرفة خلال عملية الاستدلال. و يساعد المُتَدَرِّبين على العثور على المعرفة من خلال التركيب و الألفاظ الموجودة فى لغة C++.

Agent Structure

١٠-٦-٢ تركيب المعين

شكل ١٠-٦ يعرض تركيب المعين المرشد لمُتَدَرِّبى لغة C++ و هى كالاتى :

▪ وحدة التعليم/التدريب (Tutor Module)

هذه الوحدة مسئولة عن الاستراتيجية المستخدمة فى توصيل الموضوعات و المحاضرات و تقدير التمارين المناسبة للعرض على المُتَدَرِّب/ة/ المُتَدَرِّب لحلها. عند عرض التمارين و المحاضرات يُؤخذ فى الاعتبار مستوى المُتَدَرِّب/ة/ المُتَدَرِّب. ويقوم بتغيير استراتيجيته لتساعد المُتَدَرِّب/ة/ المُتَدَرِّب بشكل أفضل.

▪ وحدة المعرفة التربوية (Pedagogical Module)

تتجزأ هذه الوحدة قرارات تربوية فى سياق التعليمات التى تحسب درجة التحكم فى الفاعلية و التفاعل المنفذة من كل من وحدة التدريب و وحدة المتدربة/المتدرب.

▪ وحدة الاستدلال (Reasoning Module)

تستخدم هذه الوحدة ثلاث أنواع من المعرفة. المعرفة التفضيلية (preference knowledge) و المعرفة التربوية (pedagogical knowledge) و معرفة المجال (domain knowledge). تحصل وحدة الاستدلال على تلك الأنواع من وحدة التدريب و وحدة المعرفة التربوية و خبير المجال على الترتيب. تستطيع هذه الوحدة أن تصل إلى تقييم عن مستوى و طريق إرشاد المتدربة/المتدرب. تقوم الوحدة باستخراج قواعد من خلال البحث و تحقق الاستدلال و تعبر عن مستوى المتدربة/المتدرب فى خمس مستويات (ممتاز و جيد جداً و جيد و مقبول و ضعيف).

▪ المُعَرِّب (Parser)

هذا البرنامج يقوم بإعراب سطور لغة C++ طبقاً للتركيبات اللفظية للغة C++ باستخدام قواعد النحو الحر (context-free grammar). يقوم المعرب بفحص مدخلات المتدربة/المتدرب لفظة تلو لفظة و يقدم تقريراً عن مستوى المتدربة/المتدرب إلى وحدة المتدربة/المتدرب.

▪ القاموس (Dictionary)

يحتوى القاموس على الكلمات المحجوزة فى لغة C و C++.

■ قاعدة معرفة المجال (Domain Knowledge Base)

مجالنا هو الطبقات و الدوال و الكلمات الأساسية و تعبيرات لغة C++. تُمثّل هذه المعرفة باستخدام شبكة معرفة لفظية من طبقات و تركيبات متعددة. كل طبقة لها كائن يصف استخداماتها و ربما يكون يأخذ عدة أشكال لتمثّل كافة الأشكال المتوقعة للطبقات و الدوال و التعبيرات. يقوم المُتَدَرِّب/المُتَدَرِّبَة بملاً الفراغ (slot) في الكائن. و لدى المرشد أمثلة لكل طبقة في عدة أشكال.

■ قاعدة المعرفة التربوية (Pedagogical Knowledge Base)

تحتوي قاعدة المعرفة التربوية على القواعد التي تحدد مستوى النصيحة و الإرشاد (حسب مستوى المُتَدَرِّب/المُتَدَرِّبَة).

■ قاعدة معرفة الأولوية (Preference Knowledge Base)

تستطيع المُتَدَرِّب/المُتَدَرِّبَة تفضيلاته و أولوياته في قاعدة معرفة الأولوية. ففي كل مرة تُعرّف المُتَدَرِّب/المُتَدَرِّبَة أحد المكونات، يتم إنشاء كائن لتخزين خبرته مع هذا المكوّن.

١٠-٦-٣ أداء المعين Agent Performance

في محيط البيئة التعليمية يستطيع المعين أن يصبح أداة مراقبة و تدريب ذكية موفرة مساعدة أساسية للمُتَدَرِّب. يستطيع المعين إنجاز المهام التالية :

- مساعدة المُتَدَرِّبين في العثور على طراز و أسلوب التعليم المناسب لهم.
- تشجيع المُتَدَرِّبين.
- مراقبة و تقييم تصرف و أداء المُتَدَرِّبين طوال الوقت.
- تنظيم المعلومات (المواضيع) و المقرر للمُتَدَرِّبين.

يقوم المعين بتسجيل خبرة المتدربين لإنشاء مساعدة شخصية لهم. و يظل المعين يراقب المتدربين ليبنى نموذج مضبوطاً لهم و يقدم التعليقات المساعدة لهم. يتفاعل المتدربون مع المعين المرشد من أجل تعلم مفاهيم جديدة أو التحقق من صحة ما تعلموه إلى هذا الحد. أثناء عملية التعلم تراقب وحدة المتدربة/المتدرب أدائهم و تحتفظ بمعلومات عما تعلموه و أدائهم خلال جلسة تعليمية.

تقدم وحدة التعليم الموضوعات حسب مستوى فهم المتدربين. نفس الطريقة تتبّع عند تقديم التمارين. مستوى الفهم يمكن تقديره من وحدة المتدربة/المتدرب.

١٠-٧ أسئلة

١. عرف Agent و Software Agent و Intelligent Agent.

٢. ما المطلوب تحديده عند تصميم المعين الذكي؟

٣. قارن بين أنواع البيانات التي يمكن أن يعمل عليها المعين.

٤. أذكر خصائص المعين الذكي.

٥. أذكر بعض تطبيقات المعين الذكي.

الفصل الحادى عشر

البحث و تعلم الآلة

Search and Machine Learning

فى هذا الفصل نتعرّف على موضوعين فى غاية الأهمية من ضمن تقنيات الذكاء الاصطناعى. الموضوع الأول يتعلّق بطرق البحث عن حل مشكلة، و هو ما يؤثّر بشكل كبير على كفاءة أى تطبيق من تطبيقات الذكاء الاصطناعى. الموضوع الثانى يتعلّق بالتعلّم فى تطبيقات الذكاء الاصطناعى و اكتساب معرفة جديدة و هى أحد أهم مميزات برامج الذكاء الاصطناعى.

1-11 تقنيات البحث Search Techniques

الكثير من مشاكل الذكاء الاصطناعى يتخللها صفة الإحباط بعدم وجود طريقة جيدة لحلها. فى الغالب يحدث ذلك، على الرغم من إمكانية إنتاج الحل جزء بعد جزء، فيفشل الحل الجزئى على حدة و يجب إعادة جزء من العمل أو كله.

1-1-11 حيز البحث/فضاء الحالات

Search Space/State Space

البحث (search) هو عملية بناء تسلسل من الأفعال يستطيع إنجاز الهدف و الوصول إلى الحل. فضاء/حيز الحالات (state space) هو تمثيل بيئة المشكلة التى نبحث لها عن حل و يتكون من مجموعة من الحالات. يتم وصف مشكلة البحث (search problem) من خلال أربعة أجزاء هى الحالة الأولية و

مجموعة العوامل أو الأفعال و دالة اختبار الهدف و تكلفة المسار. فيما يلي نوضح هذه الأجزاء.

■ **الحالة الأولية (initial state)** و هي الوضع الذي نكون عنده في البداية.

■ **مجموعة الأفعال أو العوامل (actions or operators)**. الفعل للواحد ينقلنا من حالة لأخرى، ربما تكون الحالة الهدف أو حالة أقرب إلى للحالة الهدف (أى حالة تحقق وصف الحالة الهدف). المطلوب إذن هو العثور على/الوصول إلى الحالة الهدف أو إيجاد تسلسل من الأفعال ينقلنا من الحالة الأولية إلى الحالة الهدف. بالإضافة إلى أن المشكلة قد تتطلب إيجاد أى حل أو الحل الأفضل (optimal solution). في بعض الأحيان ممكن أن نصل إلى حل و في أحيان أخرى ربما لا يوجد حل على الإطلاق.

نستخدم المصطلح **نالى (successor)** ليعنى أن الحالة S' هي الحالة التالية للحالة S إذا كان ممكناً الوصول إلى الحالة S' من الحالة S عن طريق تطبيق تسلسل من العوامل أو الأفعال. هذه العملية تتم من خلال دالة **النالى (successor function)**. فإذا كان طول التسلسل مقداره ١، يمكن الوصول إلى S' من S بتطبيق معامل واحد فقط، و يقال أن S' هي الحالة التالية مباشرة (immediate successor state). في بعض الأحيان، بدلاً من القول أن S' يوصل إليها مباشرة من S ، نقول أن S تنتج (تولد) S' . ويُطلق على تسلسل الحالات المتصلة بالأفعال اسم **المسار (path)**.

دالة التالي (successor function) ترجع بزمرة تضم معلومتين هما الحالة التالية (successor state) و الفعل (action) الذي ينقلنا من الحالة الحالية إلى الحالة التالية.

فضاء الحالات (state space)/حيز البحث (search space) هو مجموعة الحالات التي يمكن الوصول إليها من الحالة الأولية. تقليدياً، حيز البحث يمكن رسمه بيانياً كشجرة (tree) أو رسم بياني (graph) يبدأ بالحالة الأولية عند القمة و يتم توصيل كل حالة و الحالات التالية بخطوط.

- **دالة اختبار الهدف (goal test function)** و هي الدالة التي تختبر أي حالة وصلنا إليها إذا كانت الحالة الهدف. علماً بأن الحالة-الهدف (goal-state) و هي الحالة/الوضع المطلوب الوصول إليه (أي الحل).
- **دالة تكلفة المسار (path cost function)** هي الدالة تمنح كل مسار ثِقَلًا (weight) و يُطلق عليه تكلفة (cost).

الحكم على خوارزميات البحث و تقديرها

يتم تقدير جودة و نجاح خوارزميات البحث من خلال عدة عناصر :

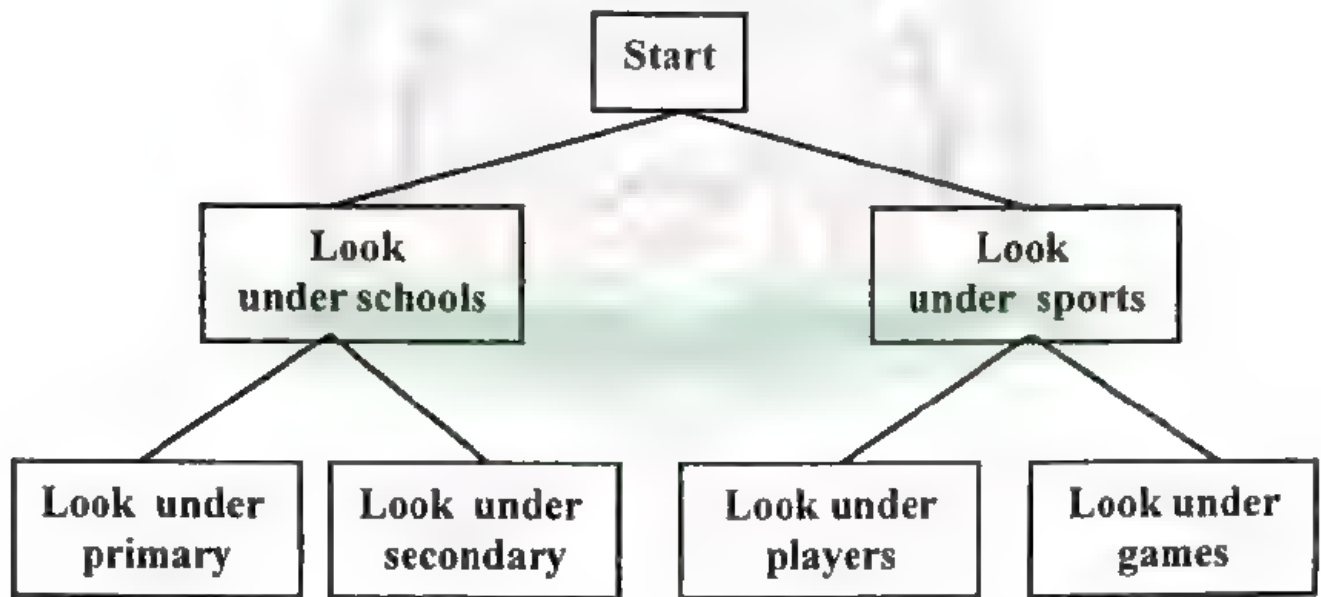
- **الكمال (Completeness).**
- **المثالية في الحل (Optimality).**
- **التعقيد الزمني (Time Complexity).**
- **تعقيد فضاء الحالات (State Space Complexity).**

١١-١-٢ مثال مشكلة بحث فى القرص عن ملف معين

Example of Disk Search Problem

فى هذا المثال مطلوب بناء المسار فى حيز/فضاء الحالات لحل المشاكل و هى العثور على ملف فى مجلد بالقرص. الحالة الأولية هى المكان أو مجموعة الأماكن التى نبدأ البحث عندها عن شىء فى القرص. وصف الحالة النهائية هو اسم الملف المطلوب العثور عليه فى القرص. يبدأ البحث فى تركيب/هيكل القرص.

الفاعل/العامل يأخذ شكل حركة من جزء (مجلد) إلى آخر. حيز/فضاء البحث هو القرص بالكامل. شكل ١-١١ يعرض مثالاً للبحث فى مجلدات القرص. و لقد اختصرنا شجرة المجلدات بالقرص إلى ما تراه فى هذا الشكل.



شكل ١-١١ : البحث عن ملف بالقرص.

Uninformed (Blind) Search

١١-١-٣ البحث الأعمى

توجد لدينا عدة تقنيات للبحث نقدم أهمها على سبيل المثال. مجموعة من تقنيات البحث تقع تحت عنوان البحث التجريبي الموجه (Heuristic Search) و ندرسها

فى الجزء التالى. بينما تقع مجموعة أخرى من تقنيات البحث تحت تصنيف البحث الأعمى (Uninformed/Blind Search) نعرضها فى هذا الجزء.

المصطلح Uninformed يعنى أن تقنيات البحث ليس لديها أى معلومات إضافية عن الحالات أكثر من الموجودة فى وصف و تعريف المشكلة. كل ما تستطيع فعله هو إنتاج تالى (successor) و تمييز الحالة الهدف عن أى حالة أخرى.

فى هذا الصنف من البحث من الممكن أن لا يوجد أى دالة تقييم مفيدة أو يكون فضاء الحالات (State Space) أو فضاء البحث (Search Space) صغير نوعاً ما بحيث تكون طريقة عادية أفضل من طريقة ماهرة. من أهم تقنيات البحث الأعمى نوعان : البحث العمقى (Depth-First Search) و البحث العرضى (Breadth-First Search)

Depth-First Search

١١-١-٣ (أ) البحث العمقى

هذا النوع من البحث يختار العقدة الأكثر عمقاً و الغير ممتدة فى شجرة البحث للاستداد. هذا النوع غير كامل و غير مثالى و به تعقيد زمنى و تعقيد فى حيز البحث/فضاء الحالات.

الخوارزم الموجود فى شكل ١١-٢ يهتم بالأماكن الممكن البحث منها باستخدام قائمة حلقات متصلة (linked List). القائمة مرتبة حسب تكلفة المسار المقترنة (estimated path cost) الذى يودى إلى الحل. الآن ليس لدينا أى قيم. تقنية البحث العمقى أو الرجوع من حيث أتى (backtracking) تتوازى مع عملية وضع حالات جديدة فى مقدمة القائمة و التحكم بها باستخدام أسلوب "من دخل مؤخراً/يخرج أولاً" (Last-In/First-Out (LIFO)). وهى تشبه معالجة المكدس/الكومة الحلقى (Linked List Stack). يعنى ذلك أنه إذا أنتجنا حالتين

$S1$ و $S2$ بتطبيق عاملين/فعلين على الحالة S ، فإن كل حالة ممكن الوصول إليها من $S1$ سيتم اختبارها قبل أى حالة يمكن الوصول إليها من $S2$ (إلا إذا كان هناك حالة يمكن الوصول إليها من كلا الحالتين $S1$ و $S2$).

Function: DFSearch

Arguments: expl - الحالة التى تُختَبَر (الحالة الجذر فى البداية)

. مُسند يرجع بها مع الحالة الهدف فقط - goal-test

.دالة ترجع بقائمة الحالات التالية فوراً لحالة ما - successors

Local variables : **Stack** - قائمة الحالات المؤكدة (المنتجة) الغير مُختَبَرَة بعد

Algorithm :

// إذا كشف goal-test أن expl هى حالة هدف اخرج //

If goal-test says that expl is a goal state then exit

Else List-expl التى ترجع بها successors عند تطبيقها على

Loop : خلال كافة أعضاء القائمة المتصلة و إلى أن تفرغ القائمة :

Try DFSearch على العضو التالى فى القائمة

If it return success // إذا رجع بالنجاح //

Then break out Loop and return the answer as

اخرج من حلقة التكرار وارجع بالإجابة كما وجدتها// Found

Endloop.

EndIf.

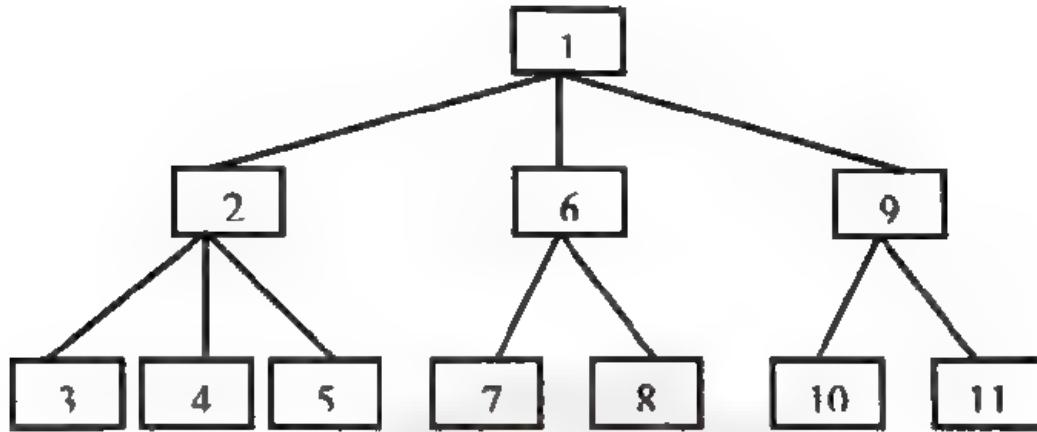
شكن ١١-٢ : خوارزم البحث العمقى (Depth-First Search)

فإذا كان هناك عدد لا نهائى يمكن الوصول إليه من $S1$ ، فلن يتم اختبار

$S2$ مُطلقاً. فى هذا الموقف، يمكن فرض نوعاً ما من الإيقاف العمقى (depth-

cutoff)، بمعنى وضع حد لأقصى طول لتسلسل تطبيق العوامل/الأفعال. سوف

نطبّق البحث العمقى فى مثال الإعراب الموجود فى الجزء التالى. شكل ١١-٣ يعرض شجرة بحث مع ترتيب اختبار الحالات موضحاً بالأرقام داخل المربعات.



شكل ١١-٣ : شجرة بحث عمقى (Depth-First Search).

Breadth-First Search

١١-١-٣ (ب) البحث العرضى

هذا النوع من البحث يختار العقدة الأقل عمقاً الغير ممتدة فى شجرة البحث للامتداد. و هو كامل و مثالى فى بعض الحالات. لكن لوجود تعقيد زمنى و تعقيد فى حيز البحث/فضاء الحالات يُصبح غير عملى فى أغلب الحالات.

الخوارزم الموجود فى شكل ١١-٤ يهتم بالأماكن الممكن البحث منها باستخدام قائمة حلقات متصلة (linked List). القائمة مُرتبة حسب تكلفة المسار المُقدّرة (estimated path cost) الذى يؤدى إلى الحل. البحث العرضى (Breadth-First Search) يوازى عملية وضع الحالات الجديدة عند نهاية القائمة و يتحكّم بها بنظام "من يدخل أولاً يخرج أولاً" (First-In/First-out (FIFO)). و هى تشبه معالجة الطابور/الرتل الحلقى (Linked List Queue). يُسمّى هذا النوع من البحث بالبحث العرضى لأنه يختبر جميع الحالات الموجودة فى نفس

مستوى الحالة الأولية (بتطبيق العوامل أو الأفعال) قبل أى حالة فى المستوى التالى.

Function: BFSearch

Arguments: expl - الحالة التى تُختَبَر (الحالة الجذر فى البداية)

goal-test - مُسند يرجع بـ 1 مع الحالة الهدف فقط -

successors - دالة ترجع بقائمة الحالات التالية فوراً لحالة ما -

Local variables : Queue - قائمة الحالات المُولَّدة (المنتجة) الغير مُختَبَرَة بعد -

Algorithm :

// إذا كشف goal-test أن expl هى حالة هدف اخرج //

If goal-test says that expl is a goal state then exit

Else List expl التى ترجع بها successors عند تطبيقها على

Loop : خلال كافة أعضاء القائمة المتصلة و إلى أن تفرغ القائمة :

Try BFSearch على العضو التالى فى القائمة

If it return success // إذا رجع بالنجاح //

Then break out Loop and return the answer as

Found // اخرج من حلقة التكرار وارجع بالإجابة كما وجدتها //

Endloop.

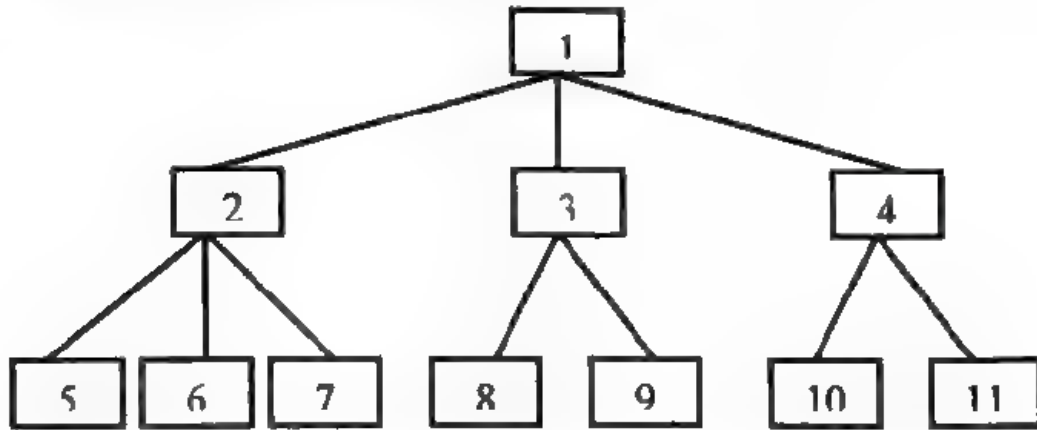
EndIf.

شكل ١١-٤ : خوارزم البحث العرضى (Breadth-First Search)

على عكس البحث العمقى، لدى البحث العرضى الخاصية أنه سوف

يستكشف أخيراً كل حالة حتى بدون إيقاف عمقى. شكل ١١-٥ يعرض شجرة

بحتم ترتيب اختبار الحالات مُوضَّحاً بالأرقام داخل المربعات.



شكل ١١-٥ : شجرة بحث عَرْضِي (Breadth-First Search).

١١-١-٤ الإعراب كمشكلة بحث

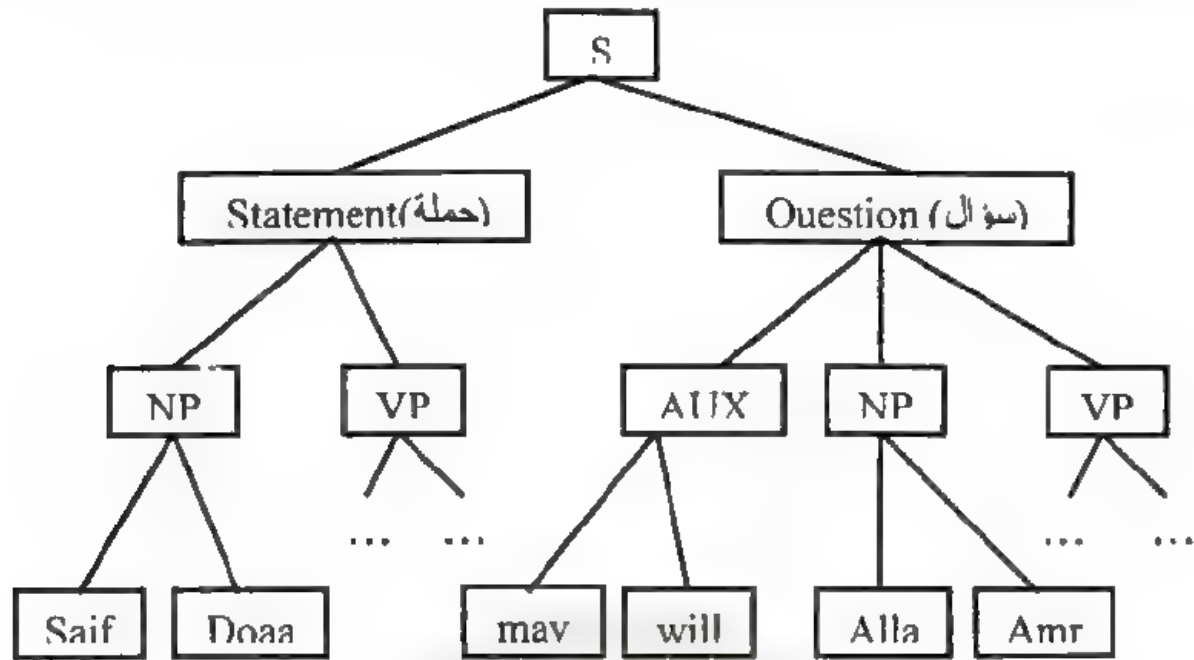
Parsing as a Search Problem

نَتَذَكَّر من الفصل الخامس قواعد النحو المستخدمة فى الإعراب. للتبسيط تأمل القاعدَتَيْن التاليتَيْن (مع العلم أن القاعدة الثانية هى نفس القاعدة الأولى و لكن فى الحالة الاستفهامية) :

$$S \rightarrow NP VP$$

$$S \rightarrow AUX NP VP$$

هذه القواعد تتطابق مع شجرة الهدف التى نراها فى الشكل ١١-٦. بالطبع مع النحو الكامل تصبح الشجرة لا نهائية عند تغطية أغلب الجمل و مفرداتها فى أى لغة طبيعية. تأمل حالة برنامج المُعَرِّب (parser) كان اختياره الأول هو القاعدة الأولى (NP VP) بدلاً من القاعدة الثانية. هذا يؤدى به إلى اختيارات إضافية : ما نوع NP؟ و ما نوع VP؟ الاختيار المُهْمَل (AUX NP VP) يمكن الاعتقاد بأنه حل جزئى بديل، يجب تخزينه لأنه من الممكن أن تبدأ الجملة فعلاً بأداة استفهام. فى هذه الحالة، سيؤدى الاختيار أخيراً إلى وضع لا يكون هناك اختيار آخر.



شكل ١١-٦: شجرة الهدف لمثال من النحو.

يستخدم برنامج المُعَرِّب البحث العمقي (Depth-First Search) (أى الرجوع من حيث أتى backtracking). بالنسبة لحالات شجرات الهدف، يعنى ذلك أن الحل الجزئى التالى الذى يستكشفه برنامج المعرب يكون امتداداً للسابق إن وُجد. فإذا لم يكن موجوداً عاد البرنامج أدراجه من حيث أتى (إلى أحدث نقطة اختيار) و يحاول امتداد آخر.

من الممكن استخدام طرق بحث أخرى. استخدام طريقة البحث العَرْضِي (Breadth-First Search) يعنى أن جميع إمتدادات الحل الجزئى ستُفَحَص "على التوازي". بعد تكريس بعض الانتباه إلى الخيار الأول (NP VP)، سوف يتحوّل إلى اختبار الخيار الثانى (AUX NP VP) و هكذا فى جميع الاختيارات الأخرى.

١١-١-٥ البحث التجريبي الموجه

Informed (Heuristic) Search

تقنيات البحث التجريبي الموجه (Heuristic Search) تعرف معلومات أكثر عن الحالات ووصف المشكلة. تستطيع تقنيات البحث الموجه أن تقدّر ما إذا كانت حالة ليست هدف أكثر فائدة من غيرها أم لا.

تقنيات البحث تلك تتجزأ بحث محلي صافى فى حيز البحث/فضاء البحث و تقوم بتقييم و تعديل حالة حالية أو أكثر بدلاً من استكشاف المسارات من الحالة الأولية. هذه التقنيات مناسبة للمشاكل حيث تكون تكلفة المسار ليست ذات صلة و كل ما يهم هو حالة الحل نفسها.

الطريقة العامة يُطلق عليها البحث التفضيلى (Best-First Search) و هى حالة من خوارزم بحث الشجرة أو بحث الرسم البيانى حيث يتم اختيار العقدة للامتداد بناءً على دالة تقييم. تقوم دالة التقييم بتقدير المسافة بين العقدة و الهدف. بناءً على ذلك يتم اختيار العقدة ذات التقييم الأقل لى يجرى لها الامتداد لأنها أقرب إلى الهدف.

١١-٢ تَعَلُّمُ الآلَةِ **Machine Learning**

من أهم مميزات برامج و تطبيقات الذكاء الاصطناعى هى قدرة البرامج (أى آلة الحاسب) على التعلّم و اكتساب معرفة جديدة. و تتنوع أشكال التعلّم اعتماداً على عنصر قاعدة المعرفة. يجب أن يكون برنامج التعلّم قادراً على التعلّم من أنواع التدريب المتاحة فى بيئته. يجب كذلك أن يستخدم أشكال التدريب الأخرى لملاحظة الأمثلة التى تمثل المفهوم و يركّب هذا التدريب ليشكّل مفاهيم دقيقة.

ربما يستطيع المرء أن يعدّل المعلومات الموجودة فى عقله أسرع من الحاسبات الموجودة حالياً. لذلك إذا استخدمنا القدرة البشرية للتعلّم و لم تكن الطرق الاصطناعية أسرع من الطبيعية، فإننا نتوقّع أن جلب و اكتساب المعرفة يكون صعباً و ربما يتطلب حاسبات أسرع من الموجودة حالياً. فيما يلى نذكر بعض أشكال تعلّم الآلة.

١١-٢-١ التعلّم الإملائى Learning by Being Told

أسهل شكل لجلب المعرفة (Knowledge Acquisition) هو التعلّم الإملائى (Learning by Being Told). فى هذا النوع نخبر الحاسب ببساطة الحقائق و القواعد عن المهمة و كذلك القواعد العامة عن كيفية استخدام تلك القواعد.

نظام Emycin و ما شابهه يستخدم هذه التقنية. يعطى هذا النظام إجابة على السؤال المُدخل، كما يعطى تفسيراً. أساساً، يختلف رد فعل النظام عند تقديم عنصر معرفة جديد للنظام حسب طبيعة العنصر الجديد. ردود الفعل هى :

- العنصر الجديد ممكن استنتاجه بالفعل من المعرفة الحالية : النظام يرفض العنصر الجديد مع رسالة مناسبة للمستخدم.
- العنصر الجديد غير متوافق و غير متجانس مع المعرفة الحالية : إما أن يرفض النظام العنصر الجديد أو يتم تعديل المعرفة الغير متوافقة معه قبل قبول العنصر الجديد. وفى أغلب الأحيان يكون الرفض هو الاختيار.
- العنصر الجديد لا يمكن استنتاجه و متجانس مع المعرفة الحالية : يُضيف النظام العنصر الجديد إلى قاعدة المعرفة و ربما يسبب نوع من التكرار.

هكذا يستطيع المرء أن يضيف إلى قاعدة المعرفة بعض الحقائق. بعض النظم تقوم بفحص و تدقيق المعرفة الجديدة. بعض النظم لا توفر هذا الفحص للمعرفة الواردة على الإطلاق و تضع المسؤولية الكاملة على المستخدم.

١١-٢-٢ التعلّم المبني على التشابه

Similarity-Based Learning

الاستقراء (Induction) أو التعميم (Generalization) هو تتبّع الأجزاء للتوصل منها إلى حكم كلى. تعلّم التعميم هام جداً لنظم المعرفة، فهو يساعد على التغلب على هشاشة نظم المعرفة. فالنظام الذى يعمل على عدة حالات اختبار يكون نظاماً هشاً. أما النظام الذى يعمل على تعميم حالات الاختبار يصبح نظاماً قوياً.

التعلّم المبني على التشابه ((Similarity-Based Learning (SBL) هو استقراء للمفاهيم العامة، مثل وصف الطبقات (class description) و التعرف على النموذج (pattern recognizer) و غيره، من التدريب باستخدام مجموعة من الأمثلة. الوصف العام للطبقة يجب أن يحتوى على أمثلة التدريب المحددة مع تلك الطبقة و يستبعد التابع لطبقة أخرى منفصلة.

فإذا تم تمثيل الأمثلة و الفئة/الطبقة فى جمل منطقية، يجب أن يدل التعميم ضمناً على الأمثلة. فالتعميم هو عبارة عن وصف ما هو عام و مشترك عن مجموعة الأمثلة التدريبية التى تنتمى إلى مجموعة معينة.

نظراً لأن عملية تشكيل التعميم من بيانات تدريبية يُطلق عليها الاستقراء أو التعميم فإن الغرض من نظام SBL هو البحث عن التعميم. لكى يتعلّم النظام بكفاءة، يجب أن يُرشد إلى تجاه التعميمات المفيدة و البناءة. يتم هذا عن طريق تقييم النتيجة النهائية مع معيار تفضيل، موفراً للنظام المتعلّم المجموعة الصحيحة

من المصطلحات الفنية التى يعمم النظام بها مع هندسة الخصائص الأولية لتلائم الغرض الذى توضع من أجله المعرفة المتعلّمة.

١١-٢-٣ التعلّم المبني على التفسير

Explanation-Based Learning

التعلّم المبني على التفسير ((Explanation-Based Learning(EBL)) هو طريقة استدلال بالاستنتاج للحصول على التعميم الصحيح. نظم EBL تستخدم نموذج المجال مثل بديهيات العلاقات الرابطة بين خصائص العوامل إلى الأهداف.

يقوم نظام EBL بتعميم مثال واحد باستنتاج خصائصه الملائمة للغرض منه، فهو يعمم تفسير أو إثبات لماذا ينتمى المثال إلى الطبقة التى تم تعميمها عن طريقه.

الخصائص العارضة أو الطارئة لمثال ما التى ربما تسبب إرباكاً لنظام EBL يتم التخلص منها على أسس منطقية. تعميمات EBL أفضل من التخمينات المبنية على غير أساس قوى، فهي مثبتة بواسطة نظرية المجال.

١١-٢-٤ التعلّم المبني على الحالات Case-Based Learning

التعلّم المبني على الحالات ((Case-Based Learning(CBL)) يُمثل بديلاً لكل من التعلم الاستقرائى SBL و التعلم الاستنتاجى EBL من الأمثلة. فهو يُشدّد على المدرس و لا يشدد على خوارزميات التعميم.

الغرض من تعميمات التعلّم فى SBL و EBL هو العثور على توصيف مُحكّم و مُلخص لكل طبقة بحيث يمكن تصنيف الأمثلة الجديدة بكفاءة. نظام التعلّم CBL يكتسب المعرفة و يستخدمها فى فهرسة و تفسير الحالات. فالتعميم هو

طبقة/فئة تمثل خلال شبكة معرفة غنية و مفهومة بالحالات التجريبية التى يُطلق عليها Exemplars للطبقات.

يتعلم النظام المبنى على الحالات بتكديس/تراكم و فهرسة الحالات. من الواضح أن النظام الذى تغطى حالاته غالبية مجال التطبيق وكذلك يغطى النجاح و الفشل يكون أفضل من ذلك الذى يغطى القليل من المجال و يغطى النجاح فقط. من الممكن أن يودى النظام المهام التالية :

- يركز على الخصائص الهامة للمشكلة و التى أدت إلى نجاح أو فشل فى الحالات السابقة.
- عمل افتراضات و توقعات لحل المشاكل.
- تقدير أن الحل الذى فشل فى موقف سابق مماثل سوف يفشل الآن.
- إعادة استخدام الاستدلال القديم عن كيفية معالجة الفشل أو تمريره لِتَجَنُّب تَكَرُّر الأخطاء السابقة.
- جلب سهل للمعرفة، نظراً لأن جمع المعرفة الفعلية من أى نجاح أو فشل سابق يكون سهلاً.
- تفسير المفاهيم ضعيفة/سيئة التعريف و خصوصاً هؤلاء المتكونين من حالات بدلاً من المفاهيم مثل المجالات القانونية و الحيوية.

١١-٢-٥ التعلم الاستقرائى (تحت إشراف)

Inductive (Supervised) Learning

من المعروف أن التعلم يأخذ عدة أشكال اعتماداً على طبيعة أداء العنصر و على الجزء المطلوب تحسينه و التغذية العكسية (أو المراجعة أو المرتدة) المتاحة (feedback). فإذا كانت التغذية العكسية المتاحة سواء من الخبير أو من مجال

التطبيق توفر القيمة الصحيحة للأمثلة، يُطلق على مشكلة التعلّم اسم التعلّم تحت إشراف (supervised learning).

قام كل من Michalaski و Chilausky بدراسة عملية و وجدوا أن قاعدة المعرفة يمكن استخلاصها أو استقرائها من الأمثلة التى يحددها أداء الخبير بدلاً من الإخبار من الخبير (التعلّم الإملائى).

بصفة عامة تكمن الصعوبة فى النقلة من الأمثلة المحتوية على الخبرة إلى القواعد العامة التى تلخص الأمثلة و قادرة على التصرف مع الأمثلة الجديدة التى لم تظهر من قبل.

يُطلق على نوع التعلّم هذا اسم التعلّم الاستقرائى (induction learning) لأنه يتعلّم الوظيفة من الأمثلة و مُدخلاتها و مُخرجاتها. يتضمن التعلّم الاستقرائى إيجاد افتراضات متوافقة تتوافق مع الأمثلة. و المفروض اختيار الافتراضات السهلة.

من أنواع التعلّم الاستقرائى الهامة التعلّم الاستقرائى لشجرات القرار (learning decision trees) و هو أحد أشكال التعلّم الناجحة و السهلة و تعتبر أساساً لخوارزميات التعلّم الاستقرائى الأخرى. و لقد قدّم Quinlan نظاماً يتم فيه تعلّم شجرة القرار من الأمثلة.

كما نعلم أن شجرة القرار مُدخلها عبارة عن كائن أو موقف على سبيل المثال، على أن نصف المُدخل بواسطة مجموعة من الخصائص، و ترجع بالقرار (وهو قيمة المخرج المتوقعة للمُدخل).

تتجز شجرة القرار عملها و تصل إلى القرار عن طريق تنفيذ تسلسل من الاختبارات. كل نقطة/عقدة فى الشجرة تمثل اختباراً على قيمة الخاصية و القيم الموضوعية على الأفرع الخارجة من العقدة تمثل القيم الممكنة لهذا الاختبار. كل عقدة ورقة فى الشجرة تحدد القيمة التى ترجع بها (المخرج) إذا تم الوصول إلى تلك العقدة.

مثال على شجرة القرار

تأمل شجرة القرار الموجودة فى شكل ٧-١١ التى تشرح مشكل حساب راتب موظف معين. الغرض من هذه المشكلة هو تعلم تعريف للهدف اختيار طريقة حساب الراتب للموظف. عند تعريف هذا الموضوع كمسكلة نعلم ينبغى تحديد الخصائص التى نعتبرها متغيرات تصف الأمثلة فى هذا المجال.



شكل ٧-١١ : شجرة قرار حساب راتب موظف.

كما نرى فى شكل ١١-٧ فإن الخصائص (العقد) التى يُجرى على قيمتها الاختبار كما يلى :

- مُوظف ثابت : هل الموظف يتقاضى راتب ثابت أو متغير.
- > 40 ساعة : هل الموظف ذو الأجر المتغير يعمل عدد ساعات أقل من 40 ساعة أسبوعياً.
- $= 40$ ساعة : هل الموظف ذو الأجر المتغير يعمل عدد ساعات $= 40$ ساعة أسبوعياً.

١١-٣ أسئلة

١. عرّف كلاً من Search و State Space.
٢. بين كيفية وصف مشكلة بحثية.
٣. ما المقصود بكل من : goal-test function و path-coast function فى مجال البحث.
٤. قارن بين كل مما يلى :
Blink Search, Depth-First Search, Breadth-First Search and Heuristic Search.
٥. اقترح مشكل بحث و بين كيفية استخدام أحد طرق البحث التى درستها فى حلها.
٦. ما المقصود بكل مما يلى :

Machine Learning, Learning by Being Told,
Similarity-Base Learning, Explanation-Based Learning,
And Case-Based Learning.

الفصل الثاني عشر

التعرف على الكلام و توليده

Speech Recognition and Synthesis

يحاول علماء الذكاء الاصطناعي مضاهاة حواس الإنسان في برامج الذكاء الاصطناعي. التركيز في أغلب أبحاث و برامج الذكاء الاصطناعي كان منصّباً على الفهم و التفكير. يوجد جانب من تطبيقات الذكاء الاصطناعي يهتم بالتعرف على الكلام (أى سماع الصوت) و جانب يهتم بتوليد و إنتاج الكلام (أى إصدار الصوت و نطقه).

Preface

١-١٢ تمهيد

الاتصال بالحاسب يأخذ أكثر من أسلوب و طريقة. فنستطيع إدخال المعلومات إلى الحاسب عن طريقة لوحة المفاتيح (keyboard) و إدخال اختيارات عن طريق الفأرة (mouse) و لإدخال الصور عن طريق الماسح الضوئي (scanner). كذلك نستطيع الحصول على المعلومات من الحاسب من شاشة العرض (monitor) أو الطابعة (printer) أو الراسم الضوئي (plotter). كما أننا ندخل و نستخرج المعلومات إلى و من الحاسب عن طريق منافذ أجهزة التخزين المختلفة.

ما نتحدث عنه في هذا الفصل هو إدخال/استخراج المعلومات و البيانات إلى/من الحاسب باستخدام الصوت مباشرة. أى مخاطبة الحاسب صوتياً و إملائه

بالتعليمات أو الأوامر أو الجمل المختلفة و كذلك سماع الكلمات التي ينطقها الحاسب.

محادثة الحاسب الآلى صوتياً يحقق مكاسب كثيرة :

- سهولة التعامل دون حاجة إلى معرفة كيفية استخدام الحاسب و لغير القادرين جسمانياً على تشغيل الحاسب.
- سرعة إدخال البيانات.
- سرية إدخال المعلومات دون وسيط لغير العارفين بتشغيل الحاسب.

استخدامات محادثة الحاسب الآلى صوتياً :

- الألعاب، ما يُعطى متعة إضافية في الألعاب للكبار و الصغار.
- التنبيه و التوجيه للأفراد في بيئة التدريب و البيئة الطبيعية.
- التعليم، يزيد الوسائط المتعددة قدرات إضافية في التعليم و التدريب.
- استخدامات المعوقين بدنياً أو بصرياً.
- الأمن، باستخدام الصوت في تعريف الشخصية و فتح الأبواب.
- الصرافة، باستخدام الصوت في تعريف المتعامل مع البنوك و صرف الأموال.
- التوجيه و التحكم في الأذرع الآلية (robots) من بُعد.

Phonemes

١٢-٢ الأصوات اللغوية

لكل حرف صوت معين و يختلف هذا الصوت باختلاف موضع الحرف في الكلمة و كذلك وجوده بعد أو قبل حرف معين. يُطلق على هذا الصوت اسم الصوت اللغوي (phoneme) و هو وحدة الكلام الصغرى.

يساعدنا الصوت اللغوي (phoneme) على تمييز كلمة عن أخرى. و ليس بالضرورة أن يكون الصوت اللغوي صوتاً لحرف، بل يمكن أن يكون صوتاً لحرفين أو ثلاثة أحرف أو أكثر. نستطيع القول أن الصوت اللغوي هو صوت مقطع لغوي أى لفظة لها صوت معين يمكن أن توجد ضمن العديد من الكلمات و لها نفس الصوت سماعاً.

بناءً على ما سبق تحتوى كل لغة على عدد من الأصوات اللغوية تتألف لتنتج لنا الكلمات التى تتكوّن منها أى لغة طبيعية. تعتمد طريقة توليد الأصوات اللغوية بالحاسب الآلى على محاكاة طريقة توليدها فى الإنسان.

فى اللغة العربية مثلاً نجد أن حروف اللغة عبارة عن أصوات لغوية أى وحدات كلام صغرى (phonemes). الجدير بالذكر أن تغيير موقع الحرف ينتج عنه وحدة كلام أخرى. كذلك نعرف حروف الخفاء (إخفاء صوت الحرف و حروف الخفاء هى حروف المد الثلاثة و الهاء) و حروف الغنة (صوت فى الخيشوم، و هى صفة لازمة للنون و التتوين و الميم فى حالات السكون و الإدغام و الخفاء) ينشأ عنها جميعاً وحدات كلام صغرى مختلفة. هذا بالإضافة إلى الحروف الساكنة و المتحركة و الحروف المخفضة و المغلظة وغيرهما.

فى اللغة الإنجليزية مثلاً نجد المقاطع led و th و gh و qua و que و qui و ide و الكثير غيرها عبارة عن أصوات لغوية (phonemes). هذا بالإضافة إلى الحروف المعروفة و غيرها من وحدات الكلام الصغرى.

١٢-٣ التعرف على الكلام (سماع الصوت)

Speech Recognition

عندما نسمع أصواتاً لأناس يتكلمون بلغة غريبة علينا، فإننا لا نفهمها و لا حتى نميّر ما يقولون (أى لا نتعرف عليها). نقصد بتمييز ما يقولون هو معرفة الكلمات و الجمل التى نطقوا بها فنستطيع كتابتها أو ترديدتها دون الوصول إلى درجة الفهم. عندما نستطيع ترديد أو كتابة كلمات مسموعة من الممكن أن نقول أننا سمعنا صوتاً فعلاً أى ميّرنا ما قيل.

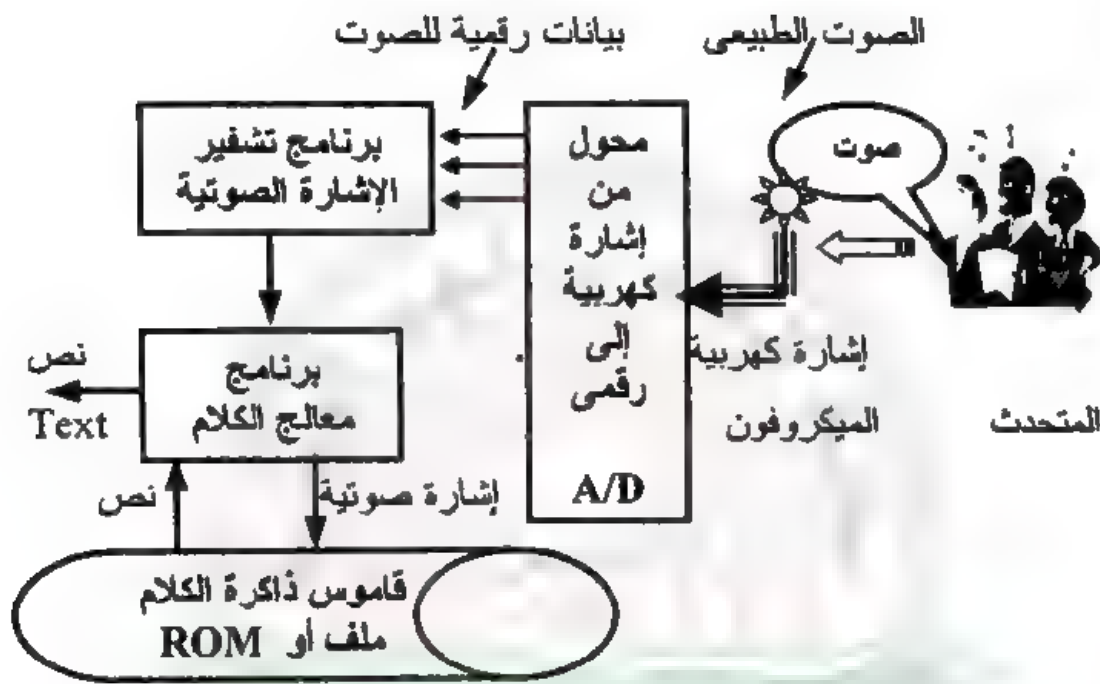
تَعْرِفُ الحاسب على الكلام أى سماع الحاسب للصوت (speech recognition) هو عملية تعرف الحاسب على الكلمات و العبارات التى ينطق بها المستخدم (speech) من خلال ميكروفون حساس و تحويلها إلى نص مكتوب (text) و من ثم الاستجابة لها أى إجراء عمليات المعالجة اللازمة مثل تخزينها كبيانات فى الذاكرة أو القرص أو تنفيذها كأوامر أو تعليمات.

عملية تعرف الحاسب على الكلام تحتاج إلى العديد من العناصر المشاركة كما نراها فى الشكل ١٢-١، هى :

- المتحدث بصوت واضح و سرعة معقولة.
- الميكروفون يحوّل الصوت إلى إشارة كهربية.
- محوّل من الإشارة الكهربائية إلى الإشارة الرقمية (Analog/Digital).
- برنامج تشفير الإشارة الصوتية الذى يقوم بقطيع الصوت المُدخّل إلى كلمات ثم مقاطع و أصوات لغوية.
- قاموس الكلام المحتوى على نماذج و مقاطع و أصوات لغوية ويحتوى الكلمات و نطقها و يختلف من برنامج لآخر. يتم تخزينه فى ملف بالقرص أو فى ذاكرة قراءة فقط (ROM). الجدير بالذكر أنه من

الممكن أن لا يتم تسجيل الصوت نفسه و لكن يتم تخزين خصائص الصوت و التي يمكن منها استعادة الصوت نفسه مرة أخرى.

■ برنامج معالج الكلام الذي يقارن الإشارات الصوتية بما هو موجود في القاموس و يستخرج الكلمات النصية المقابلة لها في القاموس. وبذلك يكون جملًا يستطيع الحاسب التعامل معه.

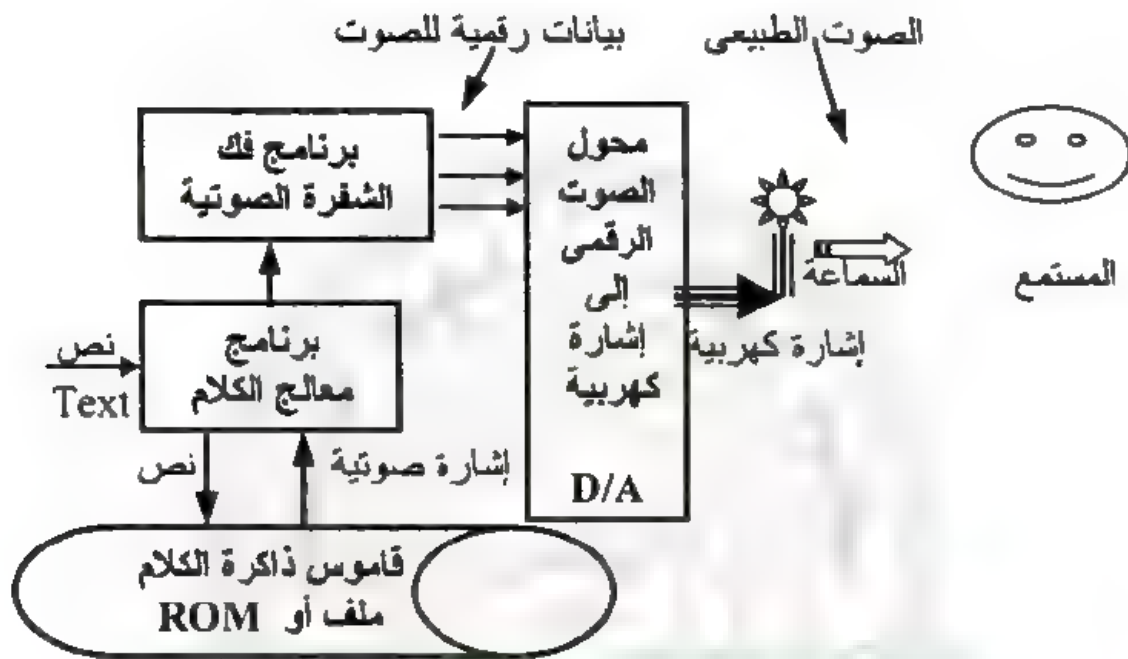


شكل ١٢-١ : عملية تحويل الأصوات إلى كلمات (التعرف على الكلام).

١٢-٤ توليد الكلام (إصدار الصوت) Speech Synthesis

توليد الكلام أو إنتاجه (speech synthesis) هي عملية تحويل النص (الكلمات المكتوبة) إلى صوت مسموع إصطناعي (كلمات منطوقة). عملية توليد الكلام من قبل الحاسب من نص إلى صوت مسموع تحتاج إلى العديد من العناصر المشاركة كما نراها في الشكل ١٢-٢، و هي :

- قاموس الكلام.
- برنامج معالج الكلام.
- برنامج فك تشفير الإشارة الصوتية.
- السماع و مكبر الصوت.
- المستمع للصوت الاصطناعي.



شكل ١٢-٢ : عملية تحويل النص المكتوب إلى صوت مسموع
(توليد الكلام).

يوجد أسلوبان لإنجاز عملية تحويل النص إلى صوت و هما :

▪ توليد الكلمات (words)

في هذه الطريقة يتم تسجيل عدد محدود من الكلمات. نتيجة لذلك، يتم نطق الكلمات بصوت واضح ولكننا لا نستطيع تخزين جميع كلمات لغة طبيعية لأن ذلك يؤدي استهلاك حيز التخزين و سرعة استخراج عالية.

■ توليد الأصوات اللغوية (phonemes)

فى هذه الطريقة يتم تخزين الإشارة الصوتية للصوت اللغوى أى وحدات الكلام الصغرى (phoneme). نتيجة لذلك، يتم تجميع الكلمات من هذه الوحدات الصغرى و يكون نطق الكلمات بصوت أقل وضوحاً ولكننا نستطيع تخزين أغلب الوحدات الصغرى و هذا لا يستهلك حيز التخزين و يؤدي إلى بطء السرعة.

١٢-٥ أسئلة

١. أذكر بعض طرق اتصال الإنسان بالحاسب.

٢. عرف كلاً من :

speech recognition, speech synthesis, phoneme

٣. ما هى مكاسب محاكاة الحاسب الآلى صوتياً؟

٤. ما هى استخدامات (تطبيقات) محاكاة الحاسب الآلى صوتياً؟

٥. أذكر أمثلة للأصوات اللغوية (phonemes) فى اللغتين العربية و الإنجليزية.

٦. ما هى العناصر المشاركة فى عملية توليد الكلام؟ مع الشرح بإيجاز دور كل منهم و علاقته بالآخرين.

٧. ما هى العناصر المشاركة فى عملية التعرف على الكلام؟ مع الشرح بإيجاز دور كل منهم و علاقته بالآخرين.



المراجع References

خالد ناصر السيد، "أصول البرمجة بلغة C++"، مكتبة الرشد، الرياض، ٢٠٠٣.

خالد ناصر السيد، "أصول تصميم قواعد البيانات و لغة SQL"، مكتبة الرشد، الرياض، ٢٠٠٣.

Abiteboul, S. and Hull, R. and Vianu, V. "Foundations of Databases", Addison-Wesley, 1995.

Allemang, D., "Combining Case-Based Reasoning and Task-Specific Architectures", IEEE EXPERT, 24-33, October 1994.

Andrews, M., "Visual C++ Object-Orient Programming", Sams Publishing, 1993.

Ashley, K., "Modeling Legal Argument : Reasoning with Cases and Hypotheticals," , Ph.D. thesis, University of Massachusetts at Amherst, 1987.

Bareiss, R., Porter, B. & Weir, C. "Protos : An Exemplar-Based Learning Apprentice", International Journal, Man-Machine Studies, 29, 549-561, 1988.

Bareiss, R., "Exemplar-Based Knowledge Acquisition, An Approach to Concept Representation, Classification, and Learning", Academic Press., Inc, 1989.

Boose, J. & Gaines, B., "The Foundations of Knowledge Acquisition", Knowledge-Based Systems Volume 4, Academic Press, 1990.

Bratko, I., "Prolog Programming for Artificial Intelligence", Addison-Wesley publishing company, 1987.

Buchanan, B., Barstow, D. & others., "Constructing an expert system", In F. Hayes-Roth, F. & Waterman, D. (Eds.), Building Expert System, 127-167., Addison-Wesley, Massachusetts, 1983.

Carbonell, J., "Machine Learning, Paradigms and methods", A Bradford Book, the MIT Press, Cambridge, Massachusetts, London, England, 1990.

Chandrasekaran, B., "Task-Structures, Knowledge Acquisition and Learning", Machine Learning, No. 4, 339-345, 1989.

Fikes, R. & Kehler, T., "The Role of Frame-Based Representation in Reasoning", Communication of the ACM, Vol. 28, No. 9, 904-919, Sept. 1985.

Forsyth, R. & Rada, R., "MACHINE LEARNING: Applications in expert systems and information retrieval", Ellis Horwood Limited, 1986.

Gaines, B., Mildred L. & Shaw G., "Eliciting Knowledge and transferring It Effectively to a Knowledge-Based System", IEEE Transaction on Knowledge and Data Engineering, Vol. 5, No. 1, 4-14, Feb. 1993.

Gruber, T. R., "The Acquisition of Strategic Knowledge", Academic Press, Inc., Based on a doctoral degree at the university of Massachusetts, 1989.

Hayes-Roth, F. & Jacobstein, N., "The State of Knowledge-Based Systems", Communication of the ACM, Vol. 37, No. 3, 27-39, Mar. 1994.

Heckerman, D., "Probabilistic Interpretations for MYCIN's Certainty Factors", In L. N. Kanal and J. F. Lemmer, Editors, Uncertainty in Artificial intelligence, Amsterdam, North-Holland, 1986.

Hennesy, D. & Hinkle, D., "Lockheed AI Center, Applying Case-Based reasoning to Autoclave Loading", IEEE EXPERT, Intelligent Systems & Their Applications, 21-26, Oct. 1992.

Kartikeyan, B., Majumder, K. & Dasgupta, A., "An Expert System for Land Cover Classification", IEEE Transactions on Geoscience and Remote Sensing, Vol. 33, No. 1, 58-66, Jan. 1995.

Kolodener, J., "Case-Based Reasoning", page 5-6, IEEE EXPERT, Oct. 1992.

Kriegsman, M. & Barletta, R., "Building a Case-Based Help Desk Application", IEEE EXPERT, 18-26, Dec. 1993.

Meng-Hiot, L., "Implementing Fuzzy Rule-Based Systems on Silicon Chips", IEEE EXPERT, 31-53, Feb. 1990.

Merrit, D., "Building Expert Systems in Prolog", Springer-Verlag Inc., NY, 1989.

Nasser, Kh., "Arabic Question Answering System (AQAS)", M.Sc. thesis, AlAzhar Univeristy, Cairo, 1991.

Nasser, Kh., Mohammad, F & Harb, H. "A Knowledge-Based Arabic Question Answering System", SIGART/ACM Press, Volume 4, Number 4, 21-31, October 1993.

Nasser, Kh., "Case-Based Reasoning Using Object-Oriented Techniques", Ph.D. thesis, Ain Shams Univeristy, Cairo, 1996.

Nasser, Kh., Badr. O. & Mohammed, F. "Nasser : Combining Featural & Action Knowledge in A Case_Based Reasoning System", The 6th International Conference on Computer Theories and Applications, Sponsored by IEEE Alexandria, Egypt, AI6: 49-56, Sept. 1996.

Nasser, Kh., Mohammed, F., Badr, O., "Nasser : Acquiring and Representing Strategic and Action Knowledge in A Case-Based Reasoning Tool", In Proceeding of the 4th IEEE International Conference On Electronics, circuits, and Systems, ICECS'97, Cairo, Egypt, Volume 2, 631-634, Dec. 1997.

Nasser, Kh., "Programming Fuzzy Logic Controller for Distillation Column", AMSE, International Conference, Kuwait Uni., Kuwait.86-96, March 2002.

Nasser, Kh., "An Intelligent Advisor Agent for C++ Trainee Students" , The Second Saudi Science Conference, King AbdaulAziz Uni., Jeddah, Saudi Arabian, March 2004.

Pearce, M., Goel, A., Kolodner J., "Case-Based Design Support : A case Study in Architectural Design" , IEEE EXPERT, 14-20, October 1992.

Quinlan, J., "Induction Of Decision Trees", Kluwer Academic Publishers, Boston - Manufactured in the Netherlands, Machine Learning 1:81-106, 1986.

Russell, S. & Norvig, P., "Artificial Intelligence, A Modern Approach", second edition, Prentice-Hall, 2003.

Simoudis, E., "Lockheed AI Center, Using Case-Based Retrieval for Customer Technical Support", IEEE EXPERT, 7-12, October 1992.

Shaw, M. & Gentry, J., "Inductive Learning for Risk Classification", IEEE EXPERT, 47-53, Feb. 1990.

Yager, R., Ovchinnikov, R. & Nguyen, H., "Fuzzy Sets and applications", Selected Papers by L. Zadeh, Wiley, NY, 1987.

YU-Huei, J. & WEI-Hsing W., "A Unified Knowledge Representation Approach in Designing an intelligent Tutor", 655-657, IEEE COMPINT, 1985.







صدر حديثاً للدكتور مهندس

خالد ناصر السيد

- مقدمة في الحاسبات والبرمجة والشبكات (مدخل إلى لغة C والإنترنت)
- أصول البرمجة بلغة C.
- أصول البرمجة بلغة C++.
- أصول تراكيب البيانات بلغتي C و C++.
- أصول تصميم قواعد البيانات ولغة SQL.
- أصول الذكاء الاصطناعي.

للمبتدئ والمتوسط والمحترف

والله الموفق والمستعان.